

THE COMMODORE 16 GAMES BOOK



JIM GREGORY

Commodore 16 Games Book

Jim Gregory

COLLINS
8 Grafton Street, London W1

Collins Professional and Technical Books
William Collins Sons & Co. Ltd
8 Grafton Street, London W1X 3LA

First published in Great Britain by
Collins Professional and Technical Books 1985

Distributed in the United States of America
by Sheridan House, Inc.

Copyright © Jim Gregory 1985

British Library Cataloguing in Publication Data
Gregory, Jim

Commodore 16 games book.
1. Computer games 2. Commodore 16(Computer)
—Programming
I. Title
794.8'028'5425 GV1469.2

ISBN 0-00-383084-5

Typeset by V & M Graphics Ltd, Aylesbury, Bucks
Printed and bound in Great Britain by
Mackays of Chatham, Kent

All rights reserved. No part of this publication may
be reproduced, stored in a retrieval system or transmitted,
in any form, or by any means, electronic, mechanical, photocopying,
recording or otherwise, without the prior permission of the
publishers.

Contents

| | |
|---|----|
| <i>Preface</i> | ix |
| <i>Introduction</i> | 1 |
| <i>Important Entry Notes</i> | 5 |
| Text Games | |
| 1 Marie Celeste: <i>A text adventure based upon discovering the legendary ghost ship.</i> | 8 |
| 2 Please Sir?: <i>A novel guessing game in which teacher has to say ‘yes’.</i> | 16 |
| 3 I Accuse: <i>A new detective game in which the criminal must be deduced from clues.</i> | 20 |
| Classic Games | |
| 4 Star Trek: <i>Using long and short range scanners engage the Klingons and fight to the death.</i> | 25 |
| 5 Rainbow Breakout: <i>Quick reflex game of breakout with bat size option.</i> | 38 |
| 6 Maze Maniac: <i>Find your way around a 3D maze as fast as you can to catch the maniac.</i> | 42 |
| Two-player Games | |
| 7 Duo Draughts: <i>The traditional game with save, load feature.</i> | 48 |
| 8 Chess Duel: <i>A beautiful board layout. Game store and replay.</i> | 55 |
| 9 Gustavus: <i>New implementation of a two-player spacial awareness game.</i> | 63 |

Dice Games

- | | | |
|----|--|----|
| 10 | Dice Derby: <i>Race along at the speed determined by the show of the dice.</i> | 66 |
| 11 | Snake Eyes: <i>Competitive dice game.</i> | 67 |
| 12 | Craps: <i>The Las Vegas favourite, including on screen betting.</i> | 68 |

Card Games

- | | | |
|----|---|----|
| 13 | Pontoon: <i>The computer banker deals the cards, the bets are placed, and one or two players try to break the bank.</i> | 76 |
| 14 | Hi-Lo: <i>Will the next card be higher or lower? Play the odds and build up a fortune in micro money.</i> | 86 |
| 15 | Concentration: <i>Two players match memory power to find pairs of cards.</i> | 93 |

Mind Games

- | | | |
|----|--|-----|
| 16 | Micro Mind: <i>Guess the colour code in ten or less, using the hints given.</i> | 100 |
| 17 | O'Grady Says: <i>Follow his instructions to move your figure – only when O'Grady says.</i> | 103 |
| 18 | Parrot: <i>Repeat the sound sequence produced by the four parrots.</i> | 108 |

Arcade Style Games

- | | | |
|----|--|-----|
| 19 | Pick Man: <i>Guide your gobbler round the maze to collect dots.</i> | 114 |
| 20 | Skippy: <i>Help your kangaroo cross the busy road and jump across the river.</i> | 122 |
| 21 | Kinkey Dong: <i>Lead the hero against falling barrels, up ladders through three different screens.</i> | 127 |

Educational Games

- | | | |
|----|--|-----|
| 22 | Word Splash: <i>Guess the word quickly to avoid getting wet.</i> | 134 |
| 23 | Sum Fun: <i>Arithmetic quiz for primary to adult.</i> | 139 |
| 24 | Today England: <i>Geographical quiz based around Britain. Option to develop further.</i> | 142 |

| | |
|---|-----|
| Utility Collection | |
| 25 Character Builder: <i>Design your own characters.</i> | 146 |
| 26 Music Monitor: <i>Great sounds and sequences to save and load.</i> | 151 |
| Final Words | 159 |
| Appendices | |
| Appendix 1: Defined Character Design Aid | 162 |
| Appendix 2: Screen Layout Planner | 163 |

Preface

This book has been produced primarily for people who enjoy games but would like to have the 'pleasure' of typing them in themselves. In this way it is possible to build up a good collection of programs for very little outlay.

Entering programs from listings is also the best way to learn the *syntax* of BASIC and to see how various results can be obtained. Later as you develop your programming skills you will be able to adapt the programs to suit your own preferences.

Finally it is hoped that you will be suitably inspired and experienced to be able to write your own programs from scratch.

Although this book can be a useful aid to learning it is not designed as a tutorial. If I had taken pages to tell you why and how each element of a program functioned then there would probably only be half as many listings! My aim and the aim of our merry band of programmers has been to ensure that each program is well worth the effort of typing in. In our efforts to ensure that the finished games are satisfactory *machine code* routines have been incorporated. The code is *POKED* into the memory from *data* and it will not be necessary to know about the wild code.

It is important to realise however that CBM BASIC, even with a little help from machine code, cannot equal a fully coded program. The compromise which has been adopted in this book is as far as it is possible to go without making each one totally machine code. In the final analysis however it is whether the game is 'playable' that matters and I hope you will agree that each game in this collection is enjoyable and playable.

This book represents a great deal of hard work and long hours from a team of people. I am pleased to be able to record my appreciation to the following Mr Micro programmers: Issi, Greg, Bootsy, David and Evan who each contributed a great deal to the

project. I am also grateful to our friends at Commodore for the loan of equipment and for their valuable technical assistance.

The other important person to have worked on the book is my wife Val who took my handwritten input and produced the finished word processed output.

Finally I am grateful to Richard Miles of Collins who commissioned this book and who ensured that despite the many possible pitfalls of publishing it has reached you, the C16 owner, for whom the entire exercise was undertaken.

Jim Gregory

Introduction

Each game featured in these pages has been specially written for the C16 and should also work with the PLUS 4. This has resulted in a range of programs covering all types of games which incorporate many useful techniques.

Each listing has been printed with a width setting of 42 columns.

After extensive testing all the listings have been produced from actual working programs. Each has been reproduced directly from the computer printout. This means that the most likely reason for a program not working is that the reader has typed it in wrong!

I have entered many programs from books and magazines, and I know from experience that at some stage you will not believe that you have got it wrong. Please read the following notes on programming carefully. They will help you to avoid the most common mistakes as you enter the programs. Follow the advice given and you will find that your microcomputer hobby can be more enjoyable and less frustrating.

Although most of the following guidelines may seem obvious, each will save you time or trouble. If you are prepared and relaxed then few errors will be made.

First program yourself

1. Set up your equipment on a desk or table, make sure that you are seated properly and try to keep your back straight. Ensure that lighting is good and that the book is supported where it can be read easily.
2. Before you start a long typing session ensure that everything is plugged in!
3. Make sure that you have a good supply of blank tapes.
4. If your blank tapes have a 'leader' on them, it is a good idea to

2 Commodore 16 Games Book

prepare them for saving by turning the hub until the brown of the tape shows.

5. To make certain everything is OK, get a long audio tape such as a C90, which will be your working tape. Next complete a 'Save and load test', e.g. type in

```
10 REM - TEST -
20 PRINT"TEST O.K."
30 GOTO 10
```

or a similar short program.

Save Test, rewind the tape, then load Test, and 'RUN' it. If this is OK, do not rewind the tape, leave it where it stopped. If all does not operate correctly then check everything and try again, until all is working. You will then be able to use this tape to store your program, either when you have finished or part way through to be continued later.

6. When you save a part version give it a title which will help you later, such as 'V1', 'V2', etc. You may prefer to use the date and time of saving, e.g. 8.15-14/10/85.

As you save a version, remember to make a note on the cassette or insert of the title and the counter number. These will enable you to retrieve the program quickly when you wish to resume entering the program or to 'debug' your completed program.

7. Always save onto tape *before* you type RUN. The microworld is full of keen folk who lost hours of work because they typed RUN before saving. (If you have saved earlier versions as directed, at least only part will be lost.)

The reasons for losing a program in this way are various. Examples are: mistyping words so that the interpreter performs a 'new' instruction, or gets into a loop from which there is no escape except by switching off. Programs which incorporate either machine code, POKE statements or System calls are most liable to 'crash', due to incorrect values being encountered by the program.

8. When a program is finished and working it is recommended that one copy is kept on a C90 back-up tape along with other programs. The main copy should be stored on a suitable short length cassette such as a C15. Only one program per side should be allowed, to enable it to be loaded easily and quickly when required. Remember to write the title on the label.

This two-tape system should help to prevent retying if a cassette is lost, damaged or develops a fault.

9. Many users become very careless about storage of cassettes and then wonder why they sometimes become faulty. If you want to keep what you've saved follow these rules:

- Identify all tapes and cases clearly.
- Keep them in cases in a rack or storage system.
- Store away from heat.
- Store away from any magnetic fields such as TV sets, computers, loudspeakers, motors, fans and very definitely telephones, all of which may damage or erase your program.
- Don't leave tapes in the deck when they are not required, and do not leave the deck switched into 'play' unless the program requires it.
- Remember to break out the plastic tabs at the back of the tape to prevent accidental erasure of the program. If ever you wish to re-record over a protected tape temporarily cover the tab hole with adhesive tape.

10. Don't spend longer than about three hours in a session. Have a break, walk about, have a shower, play a game!

Other problems

PARIS IN THE THE SPRING

Read the above out loud! You will be surprised to know that most people read it as 'Paris in the spring'. Perhaps you did too! However, it actually says 'Paris in the the spring'. The word 'the' is printed twice, but because it is not expected and because the phrase is familiar, the second 'the' is not 'seen' by the mind.

A similar problem occurs when programs are typed in from listings. You often type what you expect to see instead of what is printed. This is why the letter O is sometimes typed when it should be 'zero' (printed as Ø), or S is incorrectly typed instead of the dollar symbol (printed as \$).

A moment's lapse of concentration can lead to lines becoming jumbled as the eye jumps from one line to the next. For example:

1Ø PRINT"This is the start"

2Ø PRINT"This is next"

4 Commodore 16 Games Book

becomes:

10 PRINT“This is the next”

Look at the screen to check that your shifted letters have been correctly entered. Failure to do this produces classics such as:

10 PRINT 2 WHERE ARE THE QUOTES!2

Under no circumstances should you attempt to alter a program as it is entered. If the numbers go: 10,20,30, etc., then enter them like that. Changes are best made after a program is known to be working. This applies to everything – do not be tempted to miss out lines because you think that they may be optional. Otherwise program control could attempt to go to a line number that you have changed or deleted, and crash the program.

‘REM’ statements are remark statements and have been used to help you to understand the program. They should be entered as printed. Only when a working program has been produced and saved should you perhaps produce a copy which has no ‘REMs’. This could save a little memory, but there is no real benefit to be obtained.

Spaces are always of vital importance in a computer program. Care should be taken to get them right, to enable spaces to be counted and entered accurately. Simply align a piece of paper or card with the characters above and below, and count as the edge is moved along.

Please be particularly careful to enter each line as a complete line from the line number commands. Confusion could arise when the line includes numbers which wrap onto the next line. Do not enter these lines separately, or the computer will take it as a line number which will cause problems.

You may find it useful to place a ruler along the line you are up to. This will further reduce the risk of errors.

If you follow all the foregoing then you should not have any great problems. However, it is inevitable that some bugs will creep in. If your program fails to operate properly then the last chapter contains some helpful advice on debugging.

Important Entry Notes

Read this first

Commodore BASIC has long caused problems for those who publish listings. This is because it features special symbols to indicate things such as 'Clear home' which is a little heart character.

If we were to use a CBM printer then these symbols would be produced, but experience shows that many people have problems identifying what is what.

The programs in this book have therefore been printed using a special interface. This interface automatically prints out the special characters as easy to understand abbreviations. All such abbreviations are contained in between special brackets. These brackets indicate that the keys shown should be pressed.

The special brackets SHOULD NOT be typed in, nor should the actual letters which they contain. For example, if you see {CLR} then the Clear Screen key should be pressed. This will show on screen as the heart symbol as you enter it.

The interface also prints out the special graphic characters in a more legible form. Once again the special brackets will be printed containing either an up arrow for shift or a star for the CBM key followed by a letter. For example, if the up arrow is printed with the equals symbol then the shift key should be held down whilst the equals key is pressed. This will produce a left arrow. If a star or asterisk were printed with the equals key then the CBM logo key should be held down and equals pressed. This will produce a PI symbol.

Spaces between one set of special brackets and another set should be counted carefully by reference to the number of characters above and below.

Examples of the special bracket codes are shown below:

6 Commodore 16 Games Book

P R I N T E R C O D E S

C O L O U R S

```
{BLK}= BLACK=(CTRL+1)
{WHT}= WHITE=(CTRL+2)
{BLK}= RED=(CTRL+3)
{CYN}= CYAN=(CTRL+4)
{PUR}=PURPLE=(CTRL+5)
{GRN}= GREEN=(CTRL+6)
{BLU}= BLUE=(CTRL+7)
{YEL}=YELLOW=(CTRL+8)
{OR}=ORANGE=(CBM+1)
{BRN}=BROWN =(CBM+2)
{RVS ON}=RVS ON=(CTRL+9)
{RVS OFF}=RVS OFF=(CTRL+0)
```

CONTROL CODES

```
{CUR L}=CRSR LEFT
{CUR RT}=CURSOR RIGHT
{CUR UP}=CURSOR UP
{CUR DN}=CURSOR DOWN
{HOME}=HOME CURSOR
{CLR}=CLEAR SCREEN
O=FLASH ON =CTRL+COMMA
-=FLASH OFF=CTRL+FULL STOP
```

CHARACTER CODES

```
(*X)=CBM KEY + KEY INDICATED
(^X)=SHIFT KEY + KEY SHOWN
```

EXAMPLES

```
{^-}=SHIFT+MINUS KEY
{**}=CBM KEY+ASTERISK (MULTIPLY)
```

Remember that codes will all need to be entered in 'Quotes Mode'. This means that the first set of quotes must have been typed so that the CBM interpreter will know that a special symbol is required.

Due to the extensive use of machine code the likelihood of a *crash* is greatly increased. So do remember to *save* before you RUN. If a crash occurs it is sometimes possible to recover without loss of data.

The trick is to hold down the RUN/STOP key whilst pressing RESET. This will clear to the monitor. Type X [return] and the program should still be there.

If you have any problems refer to the *Introduction* and to the *Final Words* section.

1

Marie Celeste



An Adventure at Sea

In this text adventure you find yourself moored alongside the legendary ghost ship.

What is the secret of the *Marie Celeste*? Where are the crew?

By giving commands to the computer you can solve this marine mystery. By mixing up the text in the listing, care has been taken not to spoil your fun. In this way it is hoped that you will not be able to guess the solutions to the puzzles as they are typed in.

It is usual for adventures to require North, South, East and West as directions, but since this one has a nautical setting directions are: Forard, Aft, Starboard or Port. In case you do not know, these are equal to front of ship, rear of ship, right looking forward and left looking forward respectively.

As you move it is a good idea to draw a map. This map will help you to find your way around or identify locations that you have not yet visited.

Two-word instructions such as 'Take bucket' are accepted or the direction may be shortened to FOR, AFT, STA and POR. Adventures

often require the player to enter 'INVENTORY' to see what is being carried. In this adventure the inventory is displayed after each action. One of the enjoyable (but often frustrating) aspects of adventures is trying lots of different 'verbs' to see if they have any effect! Common verbs for adventures are: get, take, down, up, climb and unlock. Whilst you are on board, try lots of different ones. Try 'Examine cat' at the right time and see what happens.

Once you have solved the adventure invite others to play it. It is just as satisfying watching someone else make the same mistakes and perhaps have the same success as you.

```

10 REM>>>>>MARIE CELESTE<<<<<<
20 REM>>>>GREG/ISSI/JIM<<<<<<
30 GOTO1250
40 RETURN
50 PRINT"(CLR){RED}":COLOR0,1:COLOR4,1
60 PRINT"(CUR RT){CUR RT}{*Q}{^*}{*R}{^*}{^*}
^I}{^U}{^*}{^*}{^I}".
70 PRINT"(CUR RT){CUR RT}{^-}{^-}{^-}{^-}
{^-}{^-}{*R}{*R}".
80 PRINT"(CUR RT){CUR RT}{^-}{^-}{^-}{^U}
}{^*}{^I}{^U}{^*}{^I}{*R}{^U}{^*}{^I}{^-}
{^-}{^U}{^*}{^I}{^-}{^U}{^*}{^I}{^U}{^*}{^*}
I}{*Q}{^*}{^U}{^*}{^I}".
90 PRINT"(CUR RT){CUR RT}{^-}{^-}{^-}{^-}
{^-}{^-}{^-}{^-}{^-}{^-}{^-}{^-}{^-}{^-}
100 PRINT"(CUR RT){CUR RT}{^-}{^-}{^-}{^-}
{U}{^*}{*W}{^-}{^U}{^*}{*Q}{^*}{^K}{^-}
{^-}{*Q}{^*}{^K}{^-}{*Q}{^*}{^*}{^K}{^J}{^*}
{^I}{^-}{*Q}{^*}{^K}".
110 PRINT"(CUR RT){CUR RT}{^-}{^-}{^-}{^-}
{^-}{^-}{^-}{^-}{^-}{^-}{^-}{^-}{^-}{^-}
120 PRINT"(CUR RT){CUR RT}{*E}{*E}{*E}{*E}{^
J}{^*}{^K}{*E}{*E}{^J}{^*}{^K}{^J}{^*}
{^*}{^K}{^J}{^*}{^K}{*E}{^J}{^*}{^K}{^J}{^*}
{^*}{^K}{^J}{^K}{^J}{^*}{^K}".
130 PRINT"(CUR DN){YEL}":RETURN
140 REM >>>>>>> INITIALIZE<<<<<<
150 DIMD(10,7)
160 DIMOS(11)
170 DIMMM(10,4)
180 DIMO(11)

```

10 Commodore 16 Games Book

```
190 RESTORE280
200 FORN=1TO10
210 READD(N,1),D(N,2),D(N,3),D(N,4)
220 READD(N,5),D(N,6),D(N,7)
230 READM(N,1),M(N,2),M(N,3),M(N,4)
240 NEXTN
250 FORN=1TO11
260 READO$(N),O(N)
270 NEXTN:RETURN
280 DATA1,8,10,11,20,0,0
290 DATA0,0,0,0
300 DATA12,36,23,0,0,0,0
310 DATA1,1,0,0
320 DATA13,16,1,29,4,30,20
330 DATA1,0,1,0
340 DATA13,14,3,23,0,0,0
350 DATA0,1,1,0
360 DATA1,18,35,3,33,0,0
370 DATA0,1,0,0
380 DATA1,8,9,11,20,0,0
390 DATA0,0,0,0
400 DATA2,30,20,13,15,22,16
410 DATA1,0,1,1
420 DATA5,30,20,0,0,0,0
430 DATA0,0,1,0
440 DATA1,8,34,29,5,0,0
450 DATA1,0,0,1
460 DATA1,17,27,33,3,30,20
470 DATA0,0,1,0
480 DATAKNIFE,6,TARPAULIN,1,WATER ON DECK,
2,ROPE,6,NOTE,8,BUCKET,4,KEY,0
490 DATACAT,9,MOP,5,WELLIES,5,HOOKS,0
500 REM >>>>>>> INPUT <<<<<<<<<
510 PRINT:M$=""":PRINT">"":INPUTA$
520 SOUND1,800,7:M$=M$+A$
530 RETURN
540 REM >>>>>>> INVENTORY <<<<<<<<
550 PRINT"INVENTORY."
560 FORN=1TO11
570 IFO(N)=13THENPRINTO$(N)
580 NEXTN:RETURN
590 REM >>>>>>>> SPLIT <<<<<<<<<
600 V$=""":N$="""
610 FORN=1TOLEN(M$)
620 IFMID$(M$,N,1)==" "THEN640
```

```
630 NEXTN:V$=M$:RETURN
640 V$=LEFT$(M$,N-1)
650 N$=RIGHT$(M$,LEN(M$)-N)
660 RETURN
670 REM >>>>>>> COMMAND <<<<<<<
680 GOSUB510
690 GOSUB600
700 C$=LEFT$(V$,2)
710 IFC$="PO"THENGOSUB1300
720 IFC$="ST"THENGOSUB1380
730 IFC$="FC"THENGOSUB1460
740 IFC$="AF"THENGOSUB1540
750 IFC$="GE"ORC$="TA"THENGOSUB1620
760 IFC$="GO"ORC$="EN"THENGOSUB1710
770 IFC$="EX"ORC$="IN"THENGOSUB1750
780 IFC$="UN"ORC$="OP"THENGOSUB1930
790 IFC$="CU"THENGOSUB1980
800 IFC$="CL"THENGOSUB2030
810 IFC$="KI"THENGOT02100
820 IFC$="LO"THENGOSUB2150
830 IFC$="SH"ORC$="FU"ORC$="RA"THENGOSUB22
30
840 C$=LEFT$(V$,3)
850 IFC$="SWA"ORC$="MOP"THENGOSUB2250
860 IFC$="MOV"ORC$="LIF"THENGOSUB2300
870 IFV$="QUIT"THENGOT02350
880 IFV$="*"THEN900
890 PRINT"I DON'T UNDERSTAND . . ."
900 GOSUB1130:GOSUE550
910 GOTO680
920 REM >>>>>>>> START <<<<<<<<
930 GOSUB1000
940 GOSUB50
950 COSUB40
960 GOSUB150
970 GOSUE550
980 PO=6:RETURN
990 REM >>>>>>>> STRINGS <<<<<<<<
1000 DIMW$(36)
1010 RESTORE 1060
1020 FORN=1TO36
1030 READW$(N)
1040 NEXTN
1050 RETURN
```

12 Commodore 16 Games Book

```
1060 DATAI AM ,ASTERN ,FORARD ,PORT ,STARBOARD ,UP ,DOWN ,IN A ,SMALL ,LARGE
1070 DATABOAT ALONGSIDE ,THERE IS ,THE WIND ,HOWLS ,SINGS
1080 DATASOUNDS LIKE LAUGHTER. ,OUTSIDE ,INSIDE ,IT IS , THE MARIE CELESTE.
1090 DATAA ,AND ,HERE.,AND I ,SLIDE ,ACROSS ,LOCKED ,UNLOCKED ,TO ,OF ,IS
1100 DATAI SEE ,STORES.
1110 DATAGAP ,THE ,WATER ON THE DECK.,DEVIL ,BERMUDA TRIANGLE ,ATLANTIS
1120 REM >>>>>> DESCRIBE <<<<<<<
1130 REM
1140 IFPO=2ANDO(10)<>13THENPRINT"YOU SLIP
THROUGH TO THE NEXT DECK !"
1150 VOL8:SOUND2,300,10
1160 IFPO=1ANDO(2)=0THEN2410
1170 FORN=1TO7
1180 PRINTW$(D(PO,N));;
1190 NEXTN:PRINT
1200 PRINT"I CAN SEE -"
1210 FORN=1TO11
1220 IFO(N)=POTHENPRINT0$(N);";";
1230 NEXTN:PRINT:RETURN
1240 REM >>>>>> BEGIN GAME <<<<<<<
1250 GOSUB930
1260 NV$="YOU CANNOT GO THAT WAY."
1270 GOSUB1130
1280 GOTO680
1290 REM >>>>>> COMMANDS <<<<<<<
1300 V$="*"
1310 IFM(PO,4)<>0THEN1330
1320 PRINTNV$:RETURN
1330 IFM(PO,4)=1THENPO=PO-5:RETURN
1340 A=M(PO,4):A=A-1
1350 IFO(A)=13THENPRINT"YOU USE THE "0$(A)
"AND GOT PORT.":ELSEGOTO1320
1360 FORN=1TO3000:NEXTN
1370 PO=PO-5:RETURN
1380 V$="*"
1390 IFM(PO,2)<>0THEN1410
1400 PRINTNV$:RETURN
1410 IFM(PO,2)=1THENPO=PO+5:RETURN
1420 A=M(PO,2):A=A-1
```

```
1430 IFO(A)=13THENPRINT"YOU USE THE "O$(A)
"AND GO TO STARBOARD.":ELSEGOTO1400
1440 FORN=1TO3000:NEXTN
1450 PO=PO+5:RETURN
1460 V$="*"
1470 IFM(PO,1)<>0THEN1490
1480 PRINTNV$:RETURN
1490 IFM(PO,1)=1THENPC=PO+1:RETURN
1500 A=M(PO,1):A=A-1
1510 IFO(A)=13THENPRINT"YOU USE THE "O$(A)
"AND GO FORARD.":ELSEGOTO1480
1520 FORN=1TO3000:NEXTN
1530 PO=PO+1:RETURN
1540 V$="*"
1550 IFM(PO,3)<>0THEN1570
1560 PRINTNV$:RETURN
1570 IFM(PO,3)=1THENPO=PO-1:RETURN
1580 A=M(PO,3):A=A-1
1590 IFO(A)=13THENPRINT"YOU USE THE "O$(A)
" AND GO AFT.":ELSEGOTO1560
1600 FORN=1TO3000:NEXTN
1610 PO=PO-1:RETURN
1620 V$="*"
1630 FORN=1TO10
1640 IFN$=0$(N)THEN1670
1650 NEXTN
1660 PRINT"DON'T BE SILLY !":RETURN
1670 IFN=4THENPRINT"ITS TIED TOO TIGHTLY F
OR ME !":RETURN
1680 O(N)=13
1690 PRINT"YOU TAKE THE "O$(N)
1700 RETURN
1710 V$="*"
1720 IFN$<>"STORE"ANDN$<>"STORES"THENPRINT
"You CANNOT GO THERE. . .":RETURN
1730 IFM(10,4)=0THENPRINT"ITS LOCKED !":RE
TURN
1740 PO=5:RETURN
1750 V$="*"
1760 FORN=1TO11
1770 IFN$=0$(N)THEN1800
1780 NEXTN
1790 PRINT"I DON'T SEE THAT HERE.":RETURN
1800 IFN=1THENPRINT"IT'S SHARP !"
1810 IFN=2THENPRINT"IT'S GREEN !"
```

14 Commodore 16 Games Book

```
1820 IFN=3THENPRINT"IT'S WET AND SALTY !"
1830 IFN=4THENPRINT"IT'S TIED TIGHTLY."
1840 IFN=5THENPRINT"IT SAYS-":PRINT"'NO NE
ED TO GO BELOW DECKS IN THIS GAME.'"
1850 IFN=6ANDO(7)=ØTHENPRINT"IT'S RATTLING
!":O(7)=PO
1860 IFN=6ANDO(7)<>ØTHENPRINT"IT'S EMPTY !
"
1870 IFN=7THENPRINT"IT SAYS 'STORE'"
1880 IFN=8THENPRINT"IT RUNS OFF WITH A DEA
D RAT.":O(8)=Ø
1890 IFN=9THENPRINT"IT'S SOGGY !"
1900 IFN=10THENPRINT"THEY'RE SMELLY !"
1910 IFN=11THENPRINT"THEY SEEM TO BE FOR A
ROPE !"
1920 RETURN
1930 V$="*"
1940 IFPO<>10THENPRINT"I CAN'T . . .":RETURN
1950 IFO(7)<>13THENPRINT"I HAVE NO KEY !":RETURN
1960 M(PO,4)=1:PRINT"THE STORE IS OPEN."
1970 D(10,3)=28:RETURN
1980 V$="*":IFO(1)<>13THENPRINT"WITH WHAT
?":RETURN
1990 IFN$=0$(4)THEN2010
2000 PRINT"YOU SADIST !":RETURN
2010 PRINT"YOU CUT THE ROPE AND TAKE IT."
2020 O(4)=13:RETURN
2030 V$="*":IFN$<>0$(4)THENV$="BI":RETURN
2040 IFO(4)<>POANDO(4)<>13THENPRINT"WHERE'
S THE LADDER ?":RETURN
2050 PRINT"YOU CLIMB THE ROPE LADDER."
2060 IFPO=6THENPO=7:O(4)=7:RETURN
2070 IFPO=7THENPO=6:O(4)=6:RETURN
2080 IFPO=1THENPO=2:O(4)=2:RETURN
2090 RETURN
2100 PRINT"AN ALBATROSS FLIES DOWN AND KIL
LS YOU FOR BEING DESTRUCTIVE"
2110 PRINT"(CUR DN)YOU ARE DEAD !"
2120 FORN=1TO5000:NEXTN
2130 SOUND1,400,20
2140 GOT02350
2150 V$="*"
2160 PRINT"(CUR DN)EXITS ARE ;"
2170 IFM(PO,1)<>ØTHENPRINT"FORARD ";
```

```
2180 IFM(PO,3)<>0THENPRINT"AFT ";
2190 IFM(PO,2)<>0THENPRINT"STARBOARD ";
2200 IFM(PO,4)<>0THENPRINT"PORT."
2210 IFM(PO,1)=0ANDM(PO,2)=0ANDM(PO,3)=0AN
DM(PO,4)=0THENPRINT"NONE."
2220 PRINT:RETURN
2230 V$="*"
2240 PRINT"NOT NOW SAILOR !":RETURN
2250 V$="*":IFO(6)<>130R0(9)<>13THENPRINT"
WITH WHAT YOU FOOL. . .":RETURN
2260 IFPO<>2THENPRINT"NOT HERE YOU IDIOT.
.":RETURN
2270 O(3)=0:PRINT"WELL DONE ! NOW IT IS SA
FE.":M(2,3)=5
2280 D(2,1)=1:D(2,2)=2:D(2,3)=20:D(2,4)=32
:D(2,5)=21:D(2,6)=11:D(2,7)=0
2290 O(11)=2:RETURN
2300 V$="*":FORN=1TO10
2310 IFN$=0$(N)THEN2330
2320 NEXTN:PRINT"MOVE WHAT ? ! ?":RETURN
2330 PRINT"YOU MOVE IT, AND THE WIND BLOWS
IT AWAY!"
2340 O(N)=0:RETURN
2350 PRINT"(CUR DN)(CUR RT)(RVS ON)ANOTHE
R GAME ? (RVS OFF)"-
2360 GOSUB510
2370 IFM$="YES"THENRUN
2380 PRINT"(CUR DN)WELL YOU'RE PLAYING ANY
WAY !"
2390 FORN=1TO2500:NEXT
2400 RUN
2410 PRINT"SURPRISE !":PRINT"WE HID IN "W$
(21);W$(11)
2420 PRINTW$(20)"WE THOUGHT YOU WOULD"
2430 PRINT"NEVER SOLVE THE ADVENTURE !":GO
TO2350
```

2

Please Sir?



A Teaser for Everyone

This program is not quite what it seems! You may think that you are able to guess what the secret of the program is as you enter it, but you will still find the game a challenge when you play it.

The idea is that the on-screen 'teacher' selects a subject and you must then give examples which fit the subject. For example, if 'Films' was displayed then you might enter 'Please sir, is SUPERMAN one?' and teacher may say 'yes' or 'no'! You must try to decide why and then enter another example such as 'STAR WARS' to which the reply will once again be 'yes' or 'no'! The game continues until you achieve three 'yes' replies in a row, when it is assumed that you have realised why some get a 'yes' and others 'no'. You can cheat, but that most certainly would be a shallow victory.

Later, when you have mastered playing the program, you may wish to change the rules or add more subjects, and that may be a challenge in itself!

Invite a friend to list the program, and he may be surprised to find

that there is a distinct lack of data, whereupon you may either give your own reason or offer the following test of gullability:

'The program uses an advanced text compacting system based upon the mathematical characteristics of letters within English words. Thus, careful selection of parameters enables specific target answers to be 'tuned' into apparently imprecise test routines.'

Of course, you will know differently, because you typed in the lines which verify the entries! Why then, is a correct answer for the 'Birds' category: Sparrow, Chaffinch and Swallow?

```
250 REM >>> YES <<<
260 CHAR1,4,3," {CUR L}{CUR L}{CUR L}{C
UR L}YES!"
270 FORN=800TO900STEP5:SOUND1,N,2:NEXT
280 WIN=WIN+1
290 RETURN
300 REM >>> NO <<<
310 CHAR1,4,3," {CUR L}{CUR L}{CUR L}NO
!"
320 FORN=900TO800STEP-5:SOUND1,N,2:NEXT
330 WIN=0
340 RETURN
350 REM >>> START GAME <<<
360 WIN=0:G0=G0+1
370 S=INT(RND(1)*20)+1
380 CHAR1,32,1,S$(S)
390 Q$=STR$(TR):CHAR1,25,11,Q$
400 A=T(S,1):B=T(S,2):C=T(S,3)
410 RETURN
420 OK=ASC(LEFT$(A$,1))
430 IFA=2THENOK=ASC(RIGHT$(A$,1))
440 IFA=3THENOK=ASC(MID$(A$,B,1))
450 IFA=4THENOK=LEN(A$)
460 IFA=5THENOK=SP+B
470 IFS=2THENOK=ASC(LEFT$(RIGHT$(A$,2),1))
480 RETURN
490 REM >>>> INPUT <<<<<
500 A$="" : SP=0
510 CHAR1,0,16," "
520 CHAR1,0,16,A$
530 GETW$:IFW$="" THEN530
540 IFW$=" " THENSP=SP+1:GOTO580
550 IFW$=CHR$(13) THEN600
560 IFW$=CHR$(20) ANDLEN(A$)>0 THENA$=LEFT$(A$,LEN(A$)-1):GOTO590
570 IFW$<"A"ORW$>"Z" THEN530
580 A$=A$+W$:IFLEN(A$)>15 THEN500
590 SOUND1,999,5:GOTO510
600 Q=LEN(A$)
610 IFQ<10RQ>15 THEN500
620 SOUND1,400,20:SOUND1,200,20
630 GOSUB420:RE=OK
640 RETURN
650 REM >>>> LOOP <<<<<
660 GOSUB500
```

```
670 IF A$="QUIT" THEN RUN
680 IF RE=CTHENGOSUB260:ELSE GOSUB310
690 X$=STR$(WIN):CHAR1,25,11,X$
700 IF WIN=3 THEN 730
710 GOTO 660
720 REM >>>> WIN <<<<
730 SCNCLR
740 CHAR1,11,5,"CONGRATULATIONS !"
750 CHAR1,8,10,"YOU SOLVED THE PUZZLE !"
760 CHAR1,7,15,"PRESS 'Y' TO PLAY AGAIN."
770 GET A$:IF A$<>"Y" THEN 770:ELSE 60
780 REM >>>> SUBJECTS <<<<
790 DATA DOG,ACTOR,STAR,TV PROG,FILM,PLAY,
SONG
800 DATA MICRO,COLOUR,FLOWER,TREE,GEM,BOOK
,SPORT,BEAST,BIRD,PLACE
810 DATA FISH,NAME,CAR
820 DATA 2,1,78,1,2,82,3,2,79,5,1,2
830 DATA 1,1,83,5,0,0,5,5,6,4,1,8
840 DATA 4,1,4,2,1,80,4,1,3,2,1,68
850 DATA 5,2,4,3,3,67,3,4,83,3,3,65
860 DATA 2,1,82,3,2,65,3,2,65,4,1,6
870 RESTORE 790
880 DIM S$(20)
890 FOR N=1 TO 20
900 READ S$(N)
910 NEXT N
920 DIM T(20,3)
930 FOR N=1 TO 20
940 READ T(N,1),T(N,2),T(N,3)
950 NEXT N
960 RETURN
```

3

I Accuse



A Suspicious Listing

This is a test of deduction which will challenge all budding sleuths.

You are Inspector Croucher of CID (Computer Investigations Department) and are presently undertaking an extensive training course. This training is designed to test your courage and endurance. The course also tests your intelligence and ability to use the computer in criminal detection. Since we cannot cover the courage and endurance section in this book, we have included the computer deduction test.

The computer has the ability to generate thousands of different permutations from a set of data. In the blink of an eye it will decide on the criminal and his corroborative details, together with the statements of a witness that will conclusively point out the guilty party. You are allowed to ask for only six sets of details. This is done by entering the number corresponding to the category on the left of the screen. The computer will display them and then display the statements of key witnesses. This could be enough to identify the correct suspect, if you requested the right data!

The number of the suspect that you believe to be the criminal is entered and the accuracy of your judgement is reported back.

This program could well have educational value, but most of all it is designed to be different and fun.

```
10 REM <<<<< I ACCUSE >>>>>>
20 REM<<<<< ISSI/JIM >>>>>.
30 REM
40 GOSUB 990
50 GOSUB 420
60 GOSUB 570
70 FOR G=1 TO 6
80 GOSUB 1120
90 NEXT G
100 GOSUB 1230
110 COLOR 1,3,3:PRINT " YOU MUST NOW ENTER
THE NUMBER OF THE "
120 PRINT " SUSPECT YOU CONSIDER TO HAVE C
OMMITTED THE CRIME."
130 COLOR 1,15,2:PRINT " THE WITNESS SAID
THAT . . . . .":COLOR 1,5,3
140 RESTORE 200
150 FOR N=1 TO 10
160 READ A$
170 DUM=0:FOR M=1 TO 4:IF W(M)=N THEN DUM=
1
180 NEXT M:IF DUM=1 THEN PRINT " THE CRIMI
NAL ";A$;S$(S,N)
190 NEXT N
200 FOR N=1 TO 12
210 READ A,B:SOUND 1,A,B
220 NEXT N:GOTO 280
230 DATA WAS A ,WAS ,WAS ,,'S HAIR WAS ,,'S
HAIR WAS ,WORE A ,,"'S SHOES WERE "
240 DATA 'S EYES WERE ,,'S NOSE WAS ,,"HAD A
"
250 DATA 100,25,200,25,300,25,400,25
260 DATA 200,25,300,25,400,25,500,25
270 DATA 300,25,400,25,500,25,600,25
280 GETKEY A$:IF A$<"1" OR A$>"4" THEN SOU
ND 2,7,25:GOTO 280
290 CR=VAL(A$):SOUND 1,600,50:SOUND 1,400,
50:SOUND 1,200,50
300 GOSUB 1230
310 IF CR=S THEN GOTO 350
```

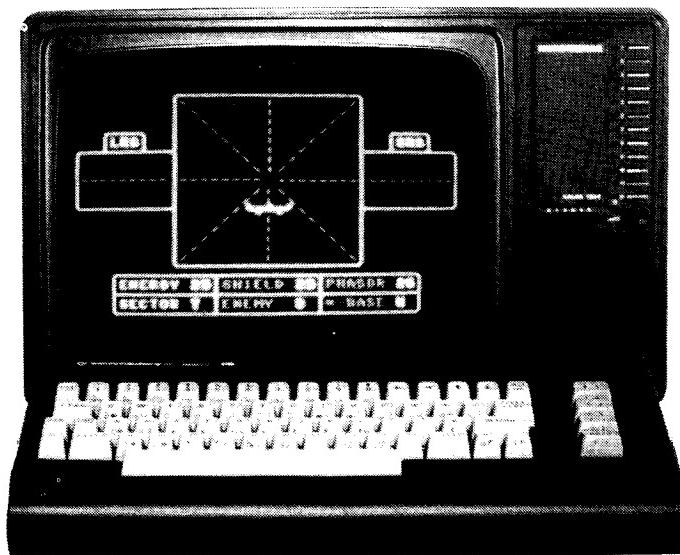
```
320 PRINT " YOU'RE PATHETIC, 'INSPECTOR' C
ROUCHER . NO ! I WILL DEMOTE YOU TO P.C."
330 PRINT:PRINT:PRINT " IT WAS SUSPECT N
UMBER ";S;" !"
340 GOTO 370
350 PRINT " CONGRATULATIONS, INSPECTOR CRO
UCHER !"
360 PRINT:PRINT "YOU CORRECTLY DEDUCED THE
CRIMINAL FROM THE CLUES GIVEN !"
370 PRINT:PRINT:PRINT " PRESS 'SPACE' T
O REPEAT GAME."
380 GETKEY A$:IF A$<>" " THEN GOTO 380
390 SOUND 1,100,50:GOTO 50
400 REM >>>>>> SCREEN <<<<<<<
410 REM
420 COLOR 0,1:COLOR 4,1:COLOR 1,3,5
430 GRAPHIC 0,1
440 PRINT "(HOME) SUSPECTS:";TAB(12);"1";T
AB(20);"2";TAB(28);"3";TAB(36);"4"
450 RESTORE 530:COLOR 1,6,2
460 PRINT:FOR N=1 TO 10
470 READ A$:PRINT A$
480 NEXT N
490 GOSUB 1230
500 COLOR 1,8,3:PRINT " CHOOSE YOUR QUESTI
ON (0-9). . . . ."
510 COLOR 1,4,4
520 RETURN
530 DATA 0-SEX,1-BUILD,2-HEIGHT,3-HAIR,4-
'COL,5-ATTIRE
540 DATA 6-SHOES,7-EYES,8-NOSE,9-SPECIAL
550 REM >>>>>> SET UP <<<<<<<
560 REM
570 S=INT(RND(1)*4)+1
580 FOR N=1 TO 4
590 W(N)=0
600 CHECK=0
610 A=INT(RND(1)*10)+1
620 FOR Q=1 TO 4
630 IF W(Q)=A THEN CHECK=1
640 NEXT Q
650 IF CHECK=1 THEN GOTO 600
660 W(N)=A
670 NEXT N
680 FOR N=1 TO 10
```

```
690 FOR Q=1 TO 4
700 A=INT(RND(1)*4)+1
710 S$(Q,N)=L$(N,A)
720 NEXT Q
730 NEXT N
740 N=W(1):GOSUB 770:N=W(3):GOSUB 770
750 N=W(2):GOSUB 870:N=W(4):GOSUB 870
760 RETURN
770 A=INT(RND(1)*4)+1
780 S$(S,N)=L$(N,A)
790 T=S+2:IF T>4 THEN T=T-4
800 S$(T,N)=L$(N,A)
810 IF A<>1 THEN B=A-1:ELSE B=4
820 T=T-2:IF T<1 THEN T=T+4
830 S$(T,N)=L$(N,B)
840 T=T-2:IF T<1 THEN T=T+4
850 S$(T,N)=L$(N,B)
860 RETURN
870 A=INT(RND(1)*4)+1
880 S$(S,N)=L$(N,A)
890 IF S<>1 THEN T=S-1:ELSE T=4
900 S$(T,N)=L$(N,A)
910 IF A<>4 THEN B=A+1:ELSE B=1
920 T=T-1:IF T=<0 THEN T=4
930 S$(T,N)=L$(N,B)
940 T=T-1:IF T=<0 THEN T=4
950 S$(T,N)=L$(N,B)
960 RETURN
970 REM >>>>>> INITIALIZE <<<<<<<
980 REM
990 DIM S$(4,10)
1000 DIM W(4)
1010 DIM L$(10,4)
1020 A=RND(-1):RESTORE 1270
1030 FOR N=1 TO 10
1040 FOR M=1 TO 4
1050 READ L$(N,M)
1060 NEXT M
1070 NEXT N
1080 VOL 8
1090 RETURN
1100 REM >>>>>>> QUESTION <<<<<<<
1110 REM
1120 GETKEY A$:IF A$<"0" OR A$>"9" THEN GO
TO 1120
```

```
1130 QW=VAL(A$)
1140 QW=QW+1
1150 SOUND 1,600-QW*50,25
1160 PRINT"(HOME)":FOR N=1 TO QW:PRINT:NEX
T N:SOUND 1,300,25
1170 FOR PE=1 TO 4
1180 PRINT TAB(PE*8+1);S$(PE,QW);
1190 NEXT PE
1200 RETURN
1210 REM >>>>>>> ERASE <<<<<<<<
1220 REM
1230 PRINT"(HOME)":FOR N=0 TO 14:PRINT:NEX
T N
1240 FOR N=15 TO 22:PRINT "
                      ":NEXT N
1250 PRINT"(HOME)":FOR N=0 TO 14:PRINT:NEX
T N
1260 RETURN
1270 DATA MAN,BOY,WOMAN,GIRL
1280 DATA FAT,THIN,SLIM,STUBBY
1290 DATA TALL,LARGE,SHORT,STOOED
1300 DATA NORMAL,SHORT,CURLY,LONG
1310 DATA BROWN,BLACK,BLONDE,GINGER
1320 DATA JEANS,SUIT,JACKET,CLOAK
1330 DATA PUMPS,BOOTS,SANDLES,CLOGS
1340 DATA BLUE,BROWN,GREEN,HAZEL
1350 DATA SMALL,LARGE,BROKEN,BENT
1360 DATA TATTOO,DAGGER,GUN,MOLE
1370 RETURN
```

4

Star Trek



The Program that Goes Boldly On

There have been versions of *Star Trek* written for just about every computer ever made. This says a lot for its attraction as a game and makes this *the* classic game.

Now you can turn your CBM into a Federation Star Ship complete with scanners and phasers. Your mission is to clear the quadrant of the evil Klingons, without using up all your energy or being vaporised.

This version incorporates a zooming star field and 'real time' action during encounters with the enemy or when docking with star bases.

There follows an extract from the 32,544 page mega-manual entitled: 'How to be Captain of a Star Ship in 4000 easy lessons':

1. *Keep calm.* Remember the most you can lose is: your own life, 20 billion Galactic Dollars' worth of hardware and the lives of everyone in the known universe.
2. *Watch where you are going!* Use the ship's eyes to reveal the nature of the universe. Pressing 'L' for Long Range Scan (LRS) will

display the position of your ship and the whereabouts of the enemy. Klingons are represented on the LRS by a circle with an arrow. This is the ancient symbol of Mars, the war bringer.

Star bases are represented by a circle above a cross. This is the ancient symbol for Venus, the bringer of love.

Those sectors which are occupied by *both* enemy and star bases have a square symbol in them. The square is an ancient symbol for a square.

3. *See what you can do?* Pressing 'S' will activate the short range scanners. This indicates the total number of enemy and star bases in the present sector.

4. *Keep an eye on your energy.* Docking with a star base will give you more. To dock press 'D' on your control panel. If there is a star base in that sector the docking sequence then requires you to steer your ship using left and right controls until the star base rests in the centre of your screen. Once docked you will receive extra units to the overall ship's energy bank and a special unit boost to your phasers and shields. Failure to dock could result in the loss of the base. So be careful out there!

5. *Zap em', Cap'n!* A Star Fleet Captain has to do what he has to do, and that means eliminating the evil menacing Klingons.

If there are Klingons around, hitting the 'B' key sounds Battle Stations and you can start zapping. First get your enemy in the centre of the screen using the cursors. When you are ready, stab the space bar, sending phaser bolts out to destroy the target.

If you are not successful quickly you will lose the element of surprise and they will start to shoot back, thus draining your shield power. Do not give in, keep zapping until the last one is cleared from the sector. Then move on to win more battles.

6. *Warp carefully.* Movement to any sector is achieved by pressing 'W' followed by a number between 1 and 5. This is equal to the warp factor or distance you wish to travel.

Next the warp direction is required. This is a number between 1 and 8. Each number corresponds to a point on a compass, starting with one at the top and counting clockwise. Note: clocks have been retained in stardate 2806 to help Captains steer their Star Ships.

7. *The semi-final frontier.* On commencement of mission you can select the level. The easy level has only 10 enemies to seek and destroy. The average level has 20 and the hard level has 30.

As each level is completed, news of your success reaches Central Klingon Control and a replacement force, increased by 10 each time, is sent out to do further battle. The ultimate challenge consists of 200

adversaries to wipe out. Are you ready Captain? Then let programming commence ...

```

10 REM <<<<<STARTREK>>>>>
20 REM <<<< DAVID/ISSI >>>
30 POKE51,255:POKE52,55:POKE55,255:POKE56,
55:PRINTCHR$(8):VOL8:REM LOWER MEMTOP
40 EN$="({HOME}{WHT}{CUR RT}{CUR RT}{CUR RT}
){CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}
{CUR RT}{CUR RT}{CUR RT}{CUR DN}{CUR DN}{CU
R DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}
{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}
{CUR DN}{CUR DN}{CUR DN}":A$="({CUR RT}{CUR
RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}
{CUR RT}{CUR RT}{CUR RT})"
50 SH$=EN$+A$
60 PH$=SH$+A$
70 SE$=EN$+"({CUR DN}{CUR DN})"
80 EM$=SE$+A$
90 AR$=EM$+A$
100 COLOR0,1:COLOR4,1:COLOR1,2:POKE1344,0
110 PRINT"(CLR)(WHT)PLEASE WAIT A MOMENT..
."
120 FORR=0TO29:READD:POKE16352+R,D:NEXT:SY
S16352
130 REM MACINE CODE TO COPY CHARSET FROM R
OM TO RAM
140 DATA160,0,185,0,208,153,0,56,185,0,209
,153,0,57,185,0,210,153,0,58,185,0,211
150 DATA153,0,59,136,208,229,96
160 REM POINT TO CHARSET NOW IN RAM
170 POKE65298,PEEK(65298)AND251:POKE65299,
PEEK(65299)AND30R56
180 REM DATA FOR SHIP SHAPE
190 FORR=0TO7:READD:POKE14336+R,D:NEXT
200 DATA7,98,242,254,242,98,7,0
210 REM DATA FOR STARS ETC
220 FORR=0TO39:READD:POKE14336+27*8+R,D:NE
XT
230 FORR=0TO7:READD:POKE14336+33*8+R,D:NEX
T
240 FORR=0TO31:READD:POKE14336+35*8+R,D:NE
XT
250 DATA,,.,.,8

```

```
260 DATA,,,,,,128
270 DATA,,,128,,,,
280 DATA128,,,,,,,
290 DATA16,,,,,,,
300 DATA1,,,,,,,
310 DATA,,,1,,,
320 DATA,,,,,,1
330 DATA15,7,13,56,198,198,198,56
340 DATA60,102,102,102,60,24,60,24
350 RESTORE3030:FORR=0TO224:READD:POKE1587
2+R,D:NEXT:REM SCROLL DATA
360 DIMS(15,2)
370 GOSUB1220
380 GOSUB1300
390 GOSUB940
400 GOSUB580
410 IFS(SEC,1)>0THENGOSUB2060
420 GOT0470
430 PRINT"(HOME){WHT}{CUR DN}{CUR DN}{CUR
DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}
)";;
440 FORR=1TO5:PRINT"(CUR RT)" "SPC(19)"":NEXT
450 PRINT"(CUR UP){CUR UP}{CUR UP}{CUR RT}#####
#####SPC(19)"]]]]]]]]]]":RETURN
460 PRINT"(HOME)":FORR=1TO39:PRINT" ";:NEXT:RETURN
470 GETA$:IFA$=="THEN550
480 IFA$="L"THENGOSUB1470
490 IFA$="S"THENGOSUB1650
500 IFA$="W"THENGOSUB1770
510 IFA$="D"THENGOSUB2130
520 IFA$="B"THENGOSUB2510
530 GOSUB580
540 IF EMEN=ALIEN THEN810
550 SYS15872
560 IF DEAD=1OR ENER <1THEN670
570 GOT0470
580 PRINT EN$" {CUR L}{CUR L}{CUR L}"ENER
590 PRINT SH$" {CUR L}{CUR L}{CUR L}"SHIE
600 PRINT PH$" {CUR L}{CUR L}{CUR L}"PHAS
610 PRINT SE$" {CUR L}{CUR L}{CUR L}"SEC
```

```

620 PRINT EM$" {CUR L}{CUR L}{CUR L}"EME
N
630 A$=" N":IFS(SEC,2)=1THEN A$=" Y"
640 PRINT AR$A$"(HOME)"
650 RETURN
660 REM >>>>>>DEAD<<<<<
670 PRINT"(CLR){CYN}{CUR DN}{CUR DN}{CUR D
N}{CUR DN}"
680 PRINTSPC(15)">BAD LUCK<(CUR DN){CUR DN
} "
690 PRINTSPC(8)"THE STARSHIP ENTERPRISE"
700 PRINT
710 PRINT" HAS BEEN DESTROYED BY THE KLING
ONS BUT{CUR DN}"
720 PRINT" ADMIRAL WELLINGTON HAS DECIDED
TO GIVE YOU{CUR DN}"
730 PRINT" JUST ONE MORE CHANCE IF YOU W
ANT TO{CUR DN}"
740 PRINTSPC(10)"GET YOUR VENGEANCE !"
750 PRINTSPC(12)"{CUR DN}{CUR DN}{CUR DN}{
YEL}ANY "CHR$(130)"KEY"CHR$(132)"TO PLAY"
760 FORT=1TO2000:NEXT
770 GETA$
780 IFA$=""THEN770
790 GOTO370
800 REM >>>>>> NEXT <<<<<<
810 PRINT"(CLR){YEL}{CUR DN}{CUR DN}{CUR D
N}{CUR DN}"
820 PRINTSPC(8)">WELL DONE CAPTAIN KIRK<(C
UR DN){CUR DN}"
830 PRINT"(GRN) NOW YOU MUST FIGHT A FIERC
ER GROUP OF{CUR DN}{CUR DN}"
840 PRINTSPC(12)"{WHT}<<<KLINGONS>>>
850 ALIEN=ALIEN+10
860 PRINTSPC(11)"{CUR DN}{CUR DN}{CUR DN}{
CUR DN}{WHT}ANY "CHR$(130)"KEY"CHR$(132)""
TO PLAY"
870 FORT=1TO2000:NEXT
880 GETA$
890 IFA$=""THEN880
900 GOTO380
910 REM <<<<< STARS >>>>>>
920 SYS15872:RETURN
930 REM <<<<< SCREEN >>>>>>
940 COLOR1,2,5
950 PRINT"(CLR){CUR DN}"SPC(10)"{^U}";:FOR

```



```

T}{CUR RT}{CUR RT}{CUR RT}{RED}*--BASE{HOME
}*
1100 PRINT"(HOME){WHT}{CUR DN}{CUR DN}";
1110 FORR=0TO7
1120 PRINTSPC(11)SPC(R)*$"SPC(7-R)"["SPC(7
-R)"]\"
1130 NEXT
1140 PRINTSPC(11)"#####{CUR RT}|||||||"
"
1150 FORR=0TO7
1160 PRINTSPC(11)SPC(7-R)"!"SPC(R)"_"SPC(R
)"^"
1170 NEXT
1180 PRINTCHR$(153)"{CUR UP}{CUR UP}{CUR U
P}{CUR UP}{CUR UP}{CUR UP}"SPC(17)"{^J}{*R
}{*E}{*R}{^K}"
1190 GOSUB580:GOSUB430
1200 RETURN
1210 REM <<<<<DIFFICULTY>>>>>
1220 PRINT"(CLR){YEL}{CUR DN}{CUR DN}{CUR
DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR RT}{CUR RT
}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR
RT}HARD EASY AVERAGE ?"
1230 GETA$
1240 IF A$="H"THEN ALIEN=20:RETURN
1250 IF A$="E"THEN ALIEN=10:RETURN
1260 IF A$="A"THEN ALIEN=15:RETURN
1270 GOTO1230
1280 PRINT"(HOME){CUR DN}{CUR DN}{CUR DN}{CUR DN
}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}";:
FORR=1TO5:PRINT"(CUR RT)      ":NEXT:PRINT"
{CUR UP}{CUR UP}{CUR UP}{CUR RT}{WHT}!!!!!
"
1290 <<<<<(INITIALISE)>>>>>
1300 FORN=1TO15:S(N,1)=0:S(N,2)=0:NEXT
1310 FORN=1TO ALIEN
1320 SEC=INT(RND(8)*15)+1
1330 IFS(SEC,1)>3THEN1320
1340 S(SEC,1)=S(SEC,1)+1
1350 NEXT
1360 FORN=1TO3
1370 PO=INT(RND(8)*15)+1
1380 IFS(PO,2)=1THEN1370
1390 S(PO,2)=1
1400 NEXT

```

```
1410 PHAS=25:ENER=25:BAD=0:SHIE=25
1420 SEC=INT(RND(8)*15)+1
1430 EMEN=0:DEAD=0
1440 SCREEN=3163:COLOUR=2139
1450 RETURN
1460 REM <<<<<LONG SCAN>>>>>
1470 FORR=0TO20:SOUND1,RND(8)*200+600,1:NE
XT
1480 PO=1:PRINT"(HOME){CUR DN}{CUR DN}{CUR
DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR D
N}{CUR DN}{CUR DN}{CUR RT}          {CUR UP
}{CUR UP}{CUR UP}{CUR UP}";;
1490 FORN=1TO3:SOUND3,1010,10:PRINT"(CUR D
N)";;
1500 FORM=1TO5:SOUND3,1020,2:PRINT"(CUR RT
)";;
1510 SYS15872
1520 IFPO=SECTHENPRINTCHR$(153) "e";:GOTO15
70
1530 IFS(PO,1)>0ANDS(PO,2)=1THENPRINTCHR$(155) "(*+)";:GOTO1570
1540 IFS(PO,1)>0THENPRINTCHR$(151) "%";:GOT
01570
1550 IFS(PO,2)=1THENPRINTCHR$(152) "&";:GOT
01570
1560 PRINT" ";
1570 FORR=1TO20:NEXT
1580 PO=PO+1:NEXTM,N:PRINT"(HOME)"
1590 FORR=1TO10:SYS15872:FORT=1TO10:NEXTT
,R
1600 GOSUB430
1610 SOUND3,1000,10
1620 ENER=ENER-5
1630 RETURN
1640 REM <<<<SHORT RANGE SCAN>>>>
1650 FORR=0TO20:SOUND1,RND(8)*200+600,1:NE
XT
1660 PRINT"(HOME){CUR DN}{CUR DN}{CUR DN}{CUR
DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}"SPC(31)CHR$(151) "%(WHT)"S(SEC,1)
1670 SOUND3,1010,20
1680 FORR=1TO400:NEXT
1690 PRINT:PRINTSPC(31)CHR$(152) "&(WHT)"S(
SEC,2)
1700 SOUND3,1020,20
```

```
1710 FORR=1TO50:SYS15872:FORT=1TO10:NEXTT,  
R  
1720 GOSUB430  
1730 SOUND3,1000,10  
1740 ENER=ENER-5  
1750 RETURN  
1760 <<<<<WARP>>>>>  
1770 PRINT"(HOME){CUR RT}{CUR RT}{CUR RT}{  
CUR RT}{YEL}WARP/DISTANCE FACTOR (1 TO 5)  
?{CUR L}{WHT}";  
1780 SOUND3,1015,5  
1790 GETA$:IFA$<"1"ORA$>"5"THEN1790:ELSEPR  
INTA$  
1800 SOUND3,1005,5  
1810 DIS=VAL(A$)  
1820 GOSUB460  
1830 PRINT"(HOME){CUR RT}{CUR RT}{CUR RT}{  
CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CY  
N}WARP DIRECTION (1 TO 8) ?{WHT}{CUR L}";  
1840 SOUND3,1015,5  
1850 GETA$:IFA$<"1"ORA$>"8"THEN1850:ELSEPR  
INTA$  
1860 SOUND3,1005,5  
1870 DATA-5,-4,1,6,5,4,-1,-6  
1880 DR=VAL(A$)  
1890 RESTORE1870  
1900 FORN=1TODR  
1910 READXY  
1920 NEXT  
1930 FORN=1TODIS  
1940 SEC=SEC+XY  
1950 IFSEC<1 THENSEC=SEC+15  
1960 IFSEC>15THENSEC=SEC-15  
1970 NEXT  
1980 FORR=800TO1020:SYS15872:SYS15872  
1990 SOUND3,R,1  
2000 NEXT  
2010 GOSUB460  
2020 IFS(SEC,1)>0THENGOSUB2060  
2030 ENER=ENER-INT(DIS*1.5)  
2040 RETURN  
2050 REM <<<<<RED ALERT>>>>>>  
2060 PRINT"(HOME){CUR RT}{CUR RT}{CUR RT}{  
CUR RT}{CUR RT}{CUR RT}{CUR RT}{YEL}"CHR$(  
130)"<<<<<(RED)RED ALERT(YEL)>>>>>>"C
```

```
HR$(132)
2070 FORR=1TO8
2080 SOUND1,600,8:SOUND2,620,8
2090 FORT=1TO100:NEXTT,R
2100 GOSUB460
2110 RETURN
2120 <<<<<<DOCKING>>>>>>
2130 IFS(SEC,2)<>0THEN2190
2140 PRINT"(HOME){CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{GRN}NO STARBASE IN
THIS SECTOR"
2150 SOUND1,300,20:SOUND2,310,20
2160 SOUND1,100,20:SOUND2,110,20
2170 FORT=1TO500:NEXT
2180 GOSUB460:RETURN
2190 Y=0:X=INT(RND(8)*17):XX=1
2200 BS=SC+X+Y*40:BC=BS-1024
2210 TS=PEEK(BS):TC=PEEK(BC)
2220 POKEBS,38:POKEBC,125
2230 GETA$:IFA$=" {CUR RT}" THENXX=XX-1
2240 IFA$=" {CUR L}" THENXX=XX+1
2250 X=X+XX:Y=Y+1
2260 IFX<0THENX=0:XX=0
2270 IFX>16THENX=16:XX=0
2280 POKEBS,TS:POKEBC,TC
2290 IFTS<>113ANDTS<>114THEN2380
2300 POKEBS,TS:POKEBC,TC
2310 PRINT"(HOME){CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{WHT}
>>>DOCKING<<<
2320 ENER=75:PHAS=75:SHIE=75
2330 FORR=1TO8:VOLR:SOUND3,400,5
2340 NEXT
2350 FORR=8TO1STEP-1:VOLR:SOUND3,400,5
2360 NEXT
2370 VOL8:SOUND3,1010,20:GOSUB460:RETURN
2380 IFY<>13THEN2430
2390 POKEBS,TS:POKEBC,TC
2400 PRINT"(HOME){CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{WHT}STARBASE OUT OF RANGE"
2410 FORR=600TO1020STEP10:SOUND3,R,2:NEXT
2420 GOSUB460:RETURN
2430 IFTS<>74ANDTS<>75THEN2480
```

```

2440 POKEBS, TS:POKEBC, TC
2450 PRINT"(HOME){CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{RED}"><<< THE STARBASE IS DE
STROYED >>>
2460 FORR=1 TO 20 TO 600 STEP 10:SOUND3,R,2:NEXT
2470 GOSUB 460:S(SEC,2)=0:RETURN
2480 GOSUB 920:SOUND1,1000,1:SOUND2,1001,1:
FORT=1 TO 20:NEXT
2490 GOTO:280
2500 REM >>>>>>>BATTLE<<<<<
2510 PRINT"(HOME){CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{WHT}"CHR$(130)">>>>>BATTLE STATIONS<<<<"CHR$(132)
2520 FORR=1 TO 4:FORR=400 TO 600 STEP 20:SOUND1
,R,2:SOUND2,R+10,2:NEXT R,RR
2530 IFS(SEC,1)<>0 THEN 2570
2540 GOSUB 460:PRINT"(HOME){CUR RT}{CUR RT}
{CUR RT}{CUR RT}{YEL}"CHR$(130)"SECTOR CLEAR"CHR$(132)
2550 FORR=1 TO 4:SOUND1,1000,2:FORT=1 TO 200:N
EXTT,R
2560 GOSUB 460:RETURN
2570 X=(INT(RND(8)*15)+1)OR1:Y=(INT(RND(8)
*15)+1)OR1:XX=1:YY=-1
2580 BS=PEEK(BS):TC=PEEK(BC)
2590 TS=PEEK(BS):TC=PEEK(BC)
2600 POKEB,37:POKEBC,123
2610 GOSUB ?28
2620 GETA$=1 IF PHAS>0 AND A$=" " THEN GOSUB 2810
2630 IF KIL>1 THEN S(SEC,1)=S(SEC,1)-1:KIL=0:
GOTO 2520
2640 IF A$=" "(CUR L)"THEN XX=1
2650 IF A$=" "(CUR RT)"THEN XX=-1
2660 IF A$=" "(CUR DN)"THEN YY=1
2670 IF A$=" "(CUR UP)"THEN YY=-1
2680 IF X=0 THEN XX=1
2690 IF X=1 THEN XX=-1
2700 IF Y=0 THEN YY=1
2710 IF Y=1 THEN YY=-1
2720 X=X+XX:Y=Y+YY
2730 IF RND(8)<.9 THEN 2780
2740 COLOR0,3,2:SOUND3,500,10:COLOR0,1,0
2750 SHIE=SHIE-1:IF SHIE>-1 THEN 2780

```

```
2760 DEAD=1
2770 PRINT"(HOME){WHT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}{CUR RT}>>>>>SHIELD
S OUT<<<<<":FORT=1TO1000:NEXT:RETURN
2780 GOSUB580
2790 FORT=1TO40:NEXT:POKEBS,TS:POKEBC,TC:G
OT02580
2800 REM <<<FIRE>>>
2810 M1=3804:M2=M1+14
2820 FORR=0T06
2830 C1=M1-1024:C2=M2-1024
2840 P=PEEK(M1):IFP<>32THEN2860
2850 POKEM1,78:POKEC1,95
2860 P=PEEK(M2):IFP<>32THEN2880
2870 POKEM2,77:POKEC2,95
2880 SOUND3,900+R*10,2
2890 M1=M1-39:M2=M2-41
2900 NEXT
2910 M1=3804:M2=M1+14
2920 FORR=0T06
2930 P=PEEK(M1):IFP<>78THEN2950
2940 POKEM1,32
2950 P=PEEK(M2):IFP<>77THEN2970
2960 POKEM2,32
2970 M1=M1-39:M2=M2-41
2980 NEXT
2990 PHAS=PHAS-1
3000 IFX=8ANDY=8THENEMEN=EMEN+1:HL=1:POKE
BS,32:COLOR0,2,7
3010 SOUND3,950,20:COLOR0,1,0
3020 RETURN
3030 DATA173,216,56,168,162,0,232,189,216,
56,202,157,216,56,232,224,7,200,243
3040 DATA152,141,223,56,173,224,56,168,162
,0,232,189,224,56,202,157,224,56
3050 DATA232,224,7,208,243,152,141,231,56,
162,7,189,224,56,74,144,2,9,126
3060 DATA157,224,56,202,16,242,173,236
3070 DATA56,74,144,2,9,128,141,234,56,173,
247,56,168,162,7,202,189,249
3080 DATA56,232,157,240,56,202,224,0,208,2
43,152,141,240,56,162,7,189,248
3090 DATA56,74,144,2,9,128,157,240,56,202,
16,242,173,255,56,168,162
3100 DATA 7,202,189,248,56,232,187,248,56,
```

202, 224
3110 DATA0, 208, 243, 152, 141, 248, 56, 173, 15, 5
7, 168, 162, 7, 202, 189, 8, 57, 232
3120 DATA157, 8, 57, 202, 224, 0, 208, 243, 152, 14
1, 8, 57, 162, 7, 189, 8, 57, 10, 144
3130 DATA2, 9, 1, 157, 8, 57, 202, 16, 242, 173, 27,
57, 10, 144, 2, 9, 1, 141, 27, 57, 173, 32
3140 DATA57, 168, 162, 0, 232
3150 DATA189, 32, 57, 202, 157, 32, 57, 232, 224, 7
, 208, 243, 152, 141, 39, 57, 162, 7
3160 DATA189, 32, 57, 10, 144, 2, 9, 1, 157, 32, 57,
202, 16, 242, 96

5

Rainbow Breakout



An Arcade Classic

This second program classic has great attraction for players of all ages. It is one of the shorter programs in the book and so may be a good first attempt for those new to programming.

At the start you are asked to select a large or small bat, then the game begins. The object of the game is to eliminate all the blocks by hitting them with the ball. Your bat is moved left or right using the cursor pad or joystick. It must be moved in advance to anticipate the path of the ball as it bounces back from the blocks.

If you lose a ball then there are three supplied before the game restarts. The score is shown and can reach very high levels in the hands of a skilful player.

```
10 REM <<<< RAINBOW ISSI/GREG >>>>>>>
20 FORN=0TO51:READA:POKE828+N,A:NEXT
30 DATA24,166,209,164,208,32,240,255,96,16
9,0,133,210,169,12,133,211,32,60,3
40 DATA152,240,8,169,40,32,100,3,136,208,2
48,138,32,100,3,177,210,133,215,96
50 DATA24,101,210,133,210,165,211,105,0,13
3,211,96
60 SC=0:LI=5:VOL8
70 PRINT"(CLR){GY 3}"
80 POKE208,12:POKE209,10:SYS828:INPUT"LARG
E/SMALL ";A$
90 S$="":LM=38
100 IFLEFT$(A$,1)="L"THEN S$="(*Y){*Y}":LM=
36
110 GOSUB250
120 CO=0
130 X=1
140 GOSUB360
150 B=23
160 N=RND(1)
170 IFN>.05THEN C=1:ELSE C=0
180 C=C+(INT(RND(1)*10)+5)
190 BD=-1:AD=-1
200 GOSUB310
210 GOSUB480
220 IFCO>300THEN110
230 GOTO200
240 REM >>>>>>>> SCREEN <<<<<<<<<
250 RESTORE280:PRINT"(CLR)":COLOR0,1:COLOR
4,1
260 FORN=1TO10:READZ:FORM=0TO38:POKE208,M:
POKE209,N:SYS828:PRINTCHR$(Z){"^Q}"
270 NEXT:NEXT
280 DATA28,28,30,30,31,31,156,156,159,159
290 RETURN
300 REM >>>>>>>> MOVEMENT <<<<<<<<<
310 GOSUB760
320 IFLE=1ANDX>0THEN GOSUB390:X=X-1:GOSUB36
0
330 IFRI=1ANDX<LMTHEN GOSUB390:X=X+1:GOSUB3
60
340 RETURN
```

```

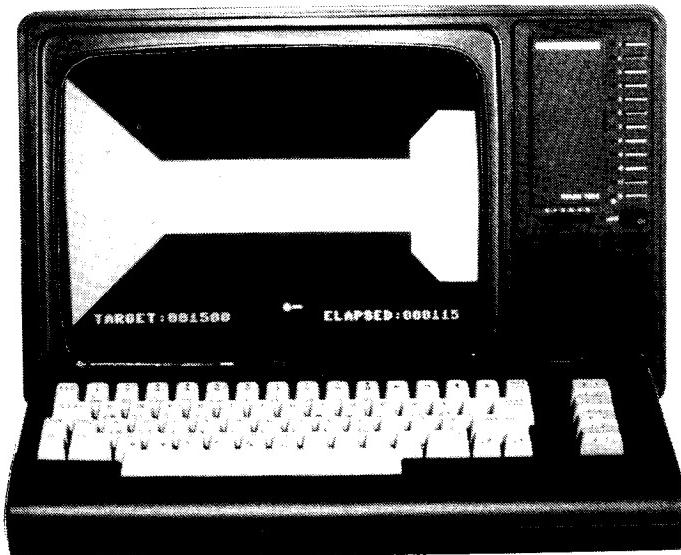
350 REM * * * DRAW * * *
360 POKE208,X:POKE209,23:SYS828:PRINT" (^O
"SS" (^P) ";
370 RETURN
380 REM * * * RUB * * *
390 POKE208,X:POKE209,23:SYS828:PRINT"   ";
400 IF$<>""THENPRINT"   ";
410 RETURN
420 REM >>>>>>> DRAW BALL <<<<<<<
430 PRINT"(WHT)";
440 POKE208,C:POKE209,B:SYS828:PRINT" (^W)-
;:RETURN
450 REM >>>>>>> RUB BALL <<<<<<<
460 POKE208,C:POKE209,B:SYS828:PRINT"   ";:R
ETURN
470 REM >>>>>> MOVE BALL <<<<<<<
480 GOSUB460:A2=C:B2=B
490 A2=A2+AD:B2=B2+BD
500 IFA2<0THENA2=0:AD=1:SOUND2,500,15
510 IFA2=39THENA2=38:AD=-1:SOUND2,500,15
520 IFB2=1THENBX=1:SOUND2,600,10:GOT0580
530 IFB2=23THEN620
540 POKE208,B2:POKE209,A2:SYS837:CH=PEEK(2
15)
550 IFCH<>81THEN570
560 POKE208,A2:POKE209,B2:SYS828:PRINT"   ";
570 IFCH=81THEN SOUND2,600,5:SC=SC+10:BX=-B
D:CO=CO+1:ELSEBX=BD
580 C=C+AD:B=B+BD
590 BD=BX
600 GOSUB430
610 RETURN
620 POKE208,23:POKE209,A2:SYS837:CH=PEEK(2
15):IFCH=32THEN670
630 BD=-1
640 N=RND(1):IFN>0.5ANDX<36THENC=C+1
650 SOUND2,100,15
660 RETURN
670 GOSUB390:LI=LI-1
680 SOUND1,75,15
690 IF LI>0THEN130
700 PRINT"(CLR)":POKE208,12:POKE209,9:SYS8
28:PRINT"CONGRATULATIONS !"
710 PRINTTAB(14)*(CUR DN)(LT RED)YOU SCORE
D : "

```

```
720 PRINTTAB(14) "CUR DN" "SC" POINTS !"
730 PRINTTAB(11) "CUR DN" O{OR}ANOTHER GAME
    ? (Y/N)"
740 GETKEYA$: IF A$ = "Y" THEN 60
750 GOTO 740
760 LE=0: RI=0: KS=PEEK(198)
770 IF KS=48 THEN LE=1
780 IF KS=51 THEN RI=1
790 RETURN
```

6

Maze Maniac



A Game with Perspective

This third classic is a 3D maze game in which you have a perspective view of the paths open to you.

The object of the game is to find your way around the maze quickly. If you do it fast enough then you will catch the maniac. If you are too slow then *he* will catch *you*!

On screen are shown the passages that are open to you. The pointer at the bottom indicates the true direction that you are facing and how far you have travelled towards the exit at bottom right.

The left or right cursor changes the direction you are facing. The up cursor takes one step into the picture each time it is pressed. The down cursor takes you back one step and leaves you facing the same direction.

Once you are successful in reaching the end of the maze then the target time is reduced and the best time must be beaten.

This game uses machine code to achieve the very fast changes to the picture.

```
10 REM >>>>> MAZE MANIAC <<{^I}<<<<<_<
20 POKE52,60:POKE54,60:POKE56,60
30 POKE51,255:POKE53,255:POKE55,255:CLR
40 SM=8:S0=SM-1:VOL 8
50 DIM M(S0,S0),MT(S0,S0)
60 GOSUB 960:TG$="001500"
70 GOSUB 850:X=0:Y=0:SP=1
80 GOSUB 180:TI$="000000"
90 SOUND 2,400,4:GOTO 470
100 GET I$:IF I$="" THEN 260
110 IP=ASC(I$):SOUND 2,800,4
120 IF IP=145 AND Y<S0 THEN IF M(X,Y+1)>0
THEN Y=Y+1:GOSUB 180
130 IF IP=17 AND Y>0 THEN IF M(X,Y-1)>0 TH
EN Y=Y-1:GOSUB 180
140 IF IP=29 THEN SP=(SP+1) AND 3:GOSUB 18
0:GOSUB 340
150 IF IP=157 THEN SP=(SP-1) AND 3:GOSUB 1
80:GOSUB 390
160 GOTO 90
170 REM >>>>> DIRECTION ARROW <<<<<
180 CHAR1,20,22,CHR$(32):CHAR1,21,23,CHR$(32)
321:CHAR1,20,24,CHR$(32)
190 CHAR1,19,23,CHR$(32):COLOR1,2
200 ON SP GOTO 220,230,240
210 CHAR1,20,22,CHR$(98):RETURN
220 CHAR1,21,23,CHR$(99):RETURN
230 CHAR1,20,24,CHR$(98):RETURN
240 CHAR1,19,23,CHR$(99):RETURN
250 REM >>>>> TIMER <<<<<<<<<<<
260 COLOR1,6,6:CHAR1,32,24,TI$
270 IF TI$<TG$ THEN 100
280 SCNCLR:COLOR1,2
290 CHAR1,4,11,"THE MAZE MANIAC HAS CAUGHT
YOU FROM"
300 CHAR1,4,13,"BEHIND. YOU WERE MUCH TOO
SLOW"
310 GOTO 830
320 RETURN
330 REM >>>>>>>> TURN RITE <<<<<<<<
340 FOR I=0 TO S0:FOR J=0 TO S0
```

```
350 MT(J,I)=M(I,S0-J)
360 NEXT J,I
370 ZZ=X:X=Y:Y=ZZ:X=S0-X:GOTO 430
380 REM >>>>>>> TURN LEFT <<<<<<<
390 :FOR I=0 TO S0:FOR J=0 TO S0
400 MT(J,S0-I)=M(I,J)
410 NEXT J,I
420 ZZ=X:X=Y:Y=ZZ:Y=S0-Y
430 FOR I=0 TO S0:FOR J=0 TO S0
440 M(J,I)=MT(J,I)
450 NEXT J,I:RETURN
460 REM >>>>> GENERATE VIEW <<<<<<<
470 FOR W=2 TO 0 STEP -1:V=Y+3-W
480 IF V=SM THEN GOSUB 600:K=W:W=0
490 IF V<SM THEN IF M(X,V)=0 THEN GOSUB 60
0:K=W:W=0
500 NEXT W:O=0:IF Y<S0 THEN IF M(X,Y+1)=2
THEN 780
510 IF X=S0 THEN FOR D=3 TO 1+K STEP -1:OC=
0:GOSUB 700:NEXT D:GOTO 550
520 V=0:FOR D=3 TO 1+K STEP -1:OC=0
530 IF M(X+1,Y+V)>0 THEN OC=1
540 GOSUB 700:V=V+1:NEXT D
550 IF X=0 THEN FOR D=6 TO 4+K STEP -1:OC=
0:GOSUB 700:NEXT D:GOTO 580
560 V=0:FOR D=6 TO 4+K STEP -1:OC=0:IF M(X
-1,Y+V)>0 THEN OC=1
570 GOSUB 700:V=V+1:NEXT D
580 GOTO 100
590 REM >>>>>> SET WALL <<<<<<<<
600 ON W GOTO 620,630
610 SYS 15646:GOTO 640
620 SYS 15624:SYS 15650:GOTO 650
630 SYS 15616:GOTO 660
640 SYS 16128:SYS 16023
650 SYS 16092:SYS 15989
660 SYS 16056:SYS 15955
670 RETURN
680 RETURN
690 REM >>>>>> SET DOOR <<<<<<<<
700 ON D GOTO 710,720,730,740,750,760
710 IF OC THEN SYS 16207:RETURN:ELSE SYS 1
6128:RETURN
720 IF OC THEN SYS 16189:RETURN:ELSE SYS 1
6092:RETURN
```

```
730 IF OC THEN SYS 16074:RETURN:ELSE SYS 1
6056:RETURN
740 IF OC THEN SYS 16251:RETURN:ELSE SYS 1
6023:RETURN
750 IF OC THEN SYS 16006:RETURN:ELSE SYS 1
5989:RETURN
760 IF OC THEN SYS 15972:RETURN:ELSE SYS 1
5955:RETURN
770 REM >>>>>> OUT <<<<<<<<<<
780 SCNCLR:COLOR1,2
790 CHAR1,4,11,"WELL DONE YOU HAVE ESCAPED
THE"
800 CHAR1,4,13,"MAZE MANIAC THIS TIME ....
"
810 CHAR1,4,15,"YOU TIME WAS "+TI$
820 TG$=TI$
830 FOR I=1 TO 5000:NEXT I:GOTO 60
840 REM >>>>> DRAW SCREEN <<<<<<<<
850 RESTORE 1360
860 FOR I=0 TO S0:FOR J=0 TO S0
870 READ M(I,J):NEXT J,I
880 COLOR 4,1:COLOR 0,1:COLOR 1,3,6
890 SCNCLR:SYS 16225:SYS 16056:SYS 15955
900 COLOR1,2:CHAR1,2,24,"TARGET:"
910 CHAR1,24,24,"ELAPSED:"
920 COLOR1,6,6:CHAR1,9,24,TG$
930 COLOR1,5,6:CHAR1,20,23,CHR$(209)
940 RETURN
950 REM >>>>>> INITIALISE <<<<<<<
960 RESTORE:AD=15616
970 READ D:IF D<0 THEN RETURN
980 POKE AD,D:AD=AD+1:GOTO 970
990 REM >>>>> PROGRAM DATA <<<<<<<
1000 REM
1010 DATA 169,163,160,12,162,17,208,6,169,
27,160,13,162,11,133,220,132,221,169
1020 DATA 160,160,32,145,220,136,208,251,7
6,37,63,162,5,208,2,162,3,169,164,160
1030 DATA 12,133,220,132,221,169,36,160,15
,133,222,132,223,160,30,169,32,132,224
1040 DATA 145,220,145,222,136,208,249,24,1
65,220,105,41,133,220,165,221,105,0
1050 DATA 133,221,56,165,222,233,39,133,22
2,165,223,233,0,133,223,164,224,136,136
1060 DATA 202,208,212,96,24,165,220,105,40
```

, 133, 220, 165, 221, 105, 0, 133, 221, 96, 165
1070 DATA 222, 145, 220, 200, 196, 224, 208, 249,
165, 223, 145, 220, 32, 99, 61, 230, 224, 160, 0
1080 DATA 202, 208, 233, 96, 165, 222, 145, 220, 2
00, 196, 224, 208, 249, 165, 223, 145, 220, 32
1090 DATA 47, 63, 230, 224, 160, 0, 202, 208, 233,
96, 165, 222, 145, 220, 136, 196, 224, 208, 247
1100 DATA 165, 223, 145, 220, 32, 99, 61, 198, 224
, 160, 4, 202, 208, 233, 96, 165, 222, 145, 220
1110 DATA 136, 196, 224, 208, 247, 165, 223, 145,
220, 32, 47, 63, 198, 224, 160, 4, 202, 208, 233
1120 DATA 96, 160, 0, 132, 224, 240, 163, 160, 0, 1
32, 224, 240, 181, 138, 168, 134, 224, 208, 199
1130 DATA 138, 168, 234, 234, 134, 224, 208, 215,
133, 222, 132, 223, 208, 224, 133, 222, 132, 223
1140 DATA 208, 224, 133, 222, 132, 223, 208, 224,
133, 222, 132, 223, 208, 224, 133, 220, 132, 221
1150 DATA 169, 160, 160, 223, 208, 222, 133, 220,
132, 221, 169, 32, 160, 32, 208, 212, 133, 220
1160 DATA 132, 221, 169, 160, 160, 105, 208, 208,
133, 220, 132, 221, 169, 32, 160, 32, 208, 198
1170 DATA 133, 220, 132, 221, 169, 160, 160, 233,
208, 194, 133, 220, 132, 221, 169, 32, 160, 32
1180 DATA 208, 184, 133, 220, 132, 221, 169, 160,
160, 95, 208, 180, 133, 220, 132, 221, 169, 32
1190 DATA 160, 32, 208, 170, 169, 0, 160, 12, 162,
4, 32, 3, 62, 169, 192, 160, 15, 162, 4, 208, 179
1200 DATA 169, 0, 160, 12, 162, 4, 32, 13, 62, 169,
192, 160, 15, 162, 4, 208, 172, 169, 164, 160
1210 DATA 12, 162, 3, 32, 3, 62, 169, 36, 160, 15, 1
62, 3, 208, 145, 169, 164, 160, 12, 162, 3, 32, 13
1220 DATA 62, 169, 36, 160, 15, 162, 3, 208, 138, 1
69, 31, 160, 13, 162, 2, 32, 3, 62, 169, 175, 160
1230 DATA 14, 162, 2, 208, 186, 169, 31, 160, 13, 3
2, 13, 62, 169, 175, 160, 14, 162, 2, 76, 33, 62
1240 DATA 169, 35, 160, 12, 162, 4, 32, 43, 62, 169
, 227, 160, 15, 162, 4, 76, 63, 62, 169, 35, 160
1250 DATA 12, 162, 4, 32, 53, 62, 169, 227, 160, 15
, 162, 4, 76, 73, 62, 169, 192, 160, 12, 162, 3
1260 DATA 32, 43, 62, 169, 64, 160, 15, 162, 3, 76,
63, 62, 169, 192, 160, 12, 162, 3, 32, 53, 62, 169
1270 DATA 64, 160, 15, 162, 3, 76, 73, 62, 169, 54,
160, 13, 162, 2, 32, 43, 62, 169, 198, 160, 14
1280 DATA 162, 2, 76, 63, 62, 169, 54, 160, 13, 162
, 2, 32, 53, 62, 169, 198, 160, 14, 162, 2, 76, 73

```
1290 DATA 62,234,32,99,61,202,208,1,96,76,  
18,61,56,165,220,233,40,133,220,165,221  
1300 DATA 233,0,133,221,96,32,238,62,169,1  
60,141,236,12,141,20,13,141,244,14,141  
1310 DATA 28,15,96,32,18,63,169,160,141,97  
,13,141,98,13,141,161,14,141,162,14  
1320 DATA 96,169,159,162,12,133,220,134,22  
1,162,17,169,160,160,40,145,220,136,208  
1330 DATA 251,32,99,61,202,208,241,96,162,  
2,76,168,62  
1340 DATA -1  
1350 REM  
1360 DATA 1,1,1,0,1,1,1,0  
1370 DATA 1,0,1,1,1,0,1,1  
1380 DATA 1,1,0,0,1,1,0,0  
1390 DATA 0,1,1,1,0,1,1,1  
1400 DATA 1,1,0,1,1,1,0,1  
1410 DATA 1,0,1,0,1,0,1,1  
1420 DATA 1,1,1,0,1,1,1,0  
1430 DATA 0,1,0,0,0,0,1,2
```

7

Duo Draughts



Computer Moderated Micro Draughts

The idea is very simple – you must get rid of all your opponent's pieces, by jumping over them. To jump over a piece there has to be an empty space on the other side.

At the start you can only move in one direction; that is, up the board if you are White or down if you are Black. To move a piece the letter for its column and the number for the row is entered. Next the destination square is given in the same way.

If it is not possible to jump over a piece then any one of your own draughts is advanced by one square to another square of the same colour. If a piece reaches the other side then it becomes a 'king' or 'crownier'. This means that it can move up and down the board and so is more powerful.

The fact that you have to take a piece when possible does not apply according to some sets of rules. Some authorities require that should an opponent's piece not be taken when possible, then the offender loses that piece. This is known as 'huffing'.

In this micro version the program will not accept an instruction

which simply moves a piece when there is a jump move available. The indicator will continue to show the same colour player until the jump sequence is complete.

A game can be saved to tape at any stage by pressing S. A previously saved game is loaded at any time by L. During a game or after loading, the step mode can be selected by pressing [shift] 3. Pressing 'Space' will then step through from the beginning. Pressing [shift] 3 will end step mode and allow the game to be played on from that position.

```

10 REM <<<<< DRAUGHTS <> ISSI >>>>>
20 REM
30 GOSUB 420
40 FOR N=1 TO 100:Z$(N,1)="00":Z$(N,2)="00"
*:NEXT
50 GOSUB 290:GOSUB 680
60 COUNT=1
70 GOSUB 1630:GOTO 70
80 GAM=0:FOR N=1 TO 8:FOR M=1 TO 8
90 IF M$(N,M)="B" OR M$(N,M)="D" THEN GAM=
1
100 NEXT:NEXT
110 IF GAM=0 THEN GOTO 180:ELSE GAM=0
120 FOR N=1 TO 8:FOR M=1 TO 8
130 IF M$(N,M)="A" OR M$(N,M)="C" THEN GAM
=1
140 NEXT:NEXT
150 IF GAM=0 THEN GOTO 220:ELSE RETURN
160 REM >>>>>>> BLACK WIN <<<<<<<<
170 REM
180 GRAPHIC 0,1:COLOR 1,4,3:CHAR 1,15,5,"B
LACK WON !"
190 GOTO 230
200 REM >>>>>>>> WHITE WIN <<<<<<<<
210 REM
220 GRAPHIC 0,1:COLOR 1,4,3:CHAR 1,15,5,"W
HITE WON !"
230 COLOR 1,3,3:CHAR 1,5,12,"DO YOU WANT A
REPLAY (Y/N) ?"
240 FOR N=1 TO 3:SOUND1,200,15:SOUND1,300,
15:SOUND1,400,15:NEXT:SOUND1,300,25
250 GETKEY A$:IF A$="Y" THEN GOSUB 1940:GO
TO 50
260 IF A$="N" THEN GOTO 50:ELSE GOTO 250
270 REM >>>>>>>> SCREEN <<<<<<<<
```

```

280 REM
290 GOSUB 680
300 COLOR 0,14,0:COLOR 4,15,2:COLOR 1,14,4
:GRAPHIC 0:GRAPHIC CLR
310 PRINT "(CLR){CUR DN) {^U}{^C}{^C}{^C}
{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^I}"
begin - 320 A$=" {-} {RVS ON} {RVS OFF} {RVS
ON} {RVS OFF} {RVS ON} {RVS OFF} {RVS
ON} {RVS OFF}{-}"
330 B$=" {-}{RVS ON} {RVS OFF} {RVS ON
} {RVS OFF} {RVS ON} {RVS OFF} {RVS ON
} {RVS OFF}{-}"
340 FOR N=1 TO 4:PRINT A$:PRINT A$:PRINT B
$:PRINT B$:NEXT N
350 PRINT " {^J}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^C}{^K
}"
360 PRINT " A B C D E F G H"
370 FOR N=1 TO 8:N$=STR$(9-N):CHAR 1,0,N*2
,N$:NEXT
380 GOSUB 750
390 RETURN
400 REM >>>>>>> CHARS <<<<<<<<
410 REM
420 FOR N=52 TO 56 STEP 2:POKE N,55:NEXT
430 RESTORE:FOR N=0 TO 30:READ A:POKE 828+
N,A:NEXT:SYS 828
440 POKE 65298,192:POKE 65299,(PEEK(65299)
AND 3)OR 56
450 FOR N=0 TO 95
460 READ A:IF N>63 THEN A=255-A
470 POKE 15136+N,A:NEXT
480 DIM M$(8,8):DIM Z$(100,2)
490 VOL 8
500 RETURN
510 DATA 160,0,132,212,132,214,169,208,133
,213,169,56,133,215,162,3
520 DATA 177,212,145,214,200,208,249,230,2
13,230,215,202,16,242,96
530 REM
540 DATA 255,255,255,255,240,143,127,31
550 DATA 255,255,255,255,15,241,254,206
560 DATA 66,80,85,85,133,240,255,255
570 DATA 194,30,226,30,241,15,255,255

```

```

580 DATA 255,255,255,255,240,128,0,0
590 DATA 255,255,255,255,15,1,0,48
600 DATA 67,80,85,85,133,240,255,255
610 DATA 194,28,224,0,1,15,255,255
620 DATA 0,1,9,86,98,86,68,128
630 DATA 0,128,144,106,70,106,26,1
640 DATA 0,1,9,87,99,118,125,255
650 DATA 0,128,144,234,198,110,58,187
660 REM >>>>> INITIALIZE <<<<<<
670 REM
680 FOR N=1 TO 8:FOR M=1 TO 8:M$(N,M)="0":NEXT:NEXT
690 FOR N=1 TO 8 STEP 2
700 M$(1,N+1)="A":M$(2,N)="A":M$(3,N+1)="A"
"
710 M$(8,N)="B":M$(7,N+1)="B":M$(6,N)="B"
720 NEXT N:RETURN
730 REM >>>>> DRAW PIECES <<<<<<
740 REM
750 FOR N=1 TO 8:FOR M=1 TO 8
760 C$=M$(N,M):IF C$="0" THEN GOTO 780
770 GOSUB 810
780 NEXT:NEXT:RETURN
790 REM >>>>>> PIECE <<<<<<<
800 REM
810 IF C$="0" THEN D$="(RVS ON) (RVS OFF)":D1$="(RVS ON) (RVS OFF)":GOTO 860
820 D$=CHR$(228)+CHR$(229):D1$=CHR$(230)+CHR$(231)
830 IF C$="A" THEN D$=CHR$(232)+CHR$(233):D1$=CHR$(234)+CHR$(235)
840 IF C$="D" THEN D$=CHR$(236)+CHR$(237)
850 IF C$="C" THEN D$=CHR$(238)+CHR$(239):D1$=CHR$(234)+CHR$(235)
860 COLOR 1,14,4:CHAR 1,M*2+1,N*2,D$:CHAR 1,M*2+1,N*2+1,D1$
870 RETURN
880 REM >>>>>>>> MOVE <<<<<<<
890 REM
900 COLOR 1,14,4:FOR N=2 TO 15:CHAR 1,25,N,"(RVS ON) (RVS OFF)":NEXT
910 COLOR 1,3,2:FOR N=6 TO 13:CHAR 1,26,N,"":NEXT
920 COLOR 1,16,4:CHAR 1,26,3," DRAUGHTS. ":"CHAR 1,26,4,"-----"

```

```

930 COLOR 1,8,4:IF PLA=1 THEN N$="WHITE":E
LSE N$="BLACK"
940 CHAR 1,29,6,N$:CHAR 1,27,8,"FROM:-":CH
AR 1,27,11,"TO :"
950 Y=8:GOSUB 980:F$=N$
960 Y=11:GOSUB 980:S$=N$
970 RETURN
980 N$=""
990 GETKEY A$:SOUND 1,500,10
1000 IF A$="Q" THEN RUN
1010 IF A$="S" THEN GOSUB 1680
1020 IF A$="L" THEN GOSUB 1800:GOTO 900
1030 IF A$="#" AND COUNT>1 THEN Z$(COUNT,1)
)=FF":GOTO 1940
1040 IF A$<"A" OR A$>"H" THEN GOTO 990
1050 N$=A$:CHAR 1,34,Y,N$
1060 GETKEY A$:SOUND 1,600,10
1070 IF A$<"1" OR A$>"9" THEN GOTO 1060
1080 N$=N$+A$:CHAR 1,34,Y,N$:RETURN
1090 REM >>>>>>>> MOVE <<<<<<<<
1100 REM
1110 GOSUB 1230
1120 M$(S1,S)=M$(F1,F):M$(F1,F)="@"
1130 IF S1=1 AND M$(1,S)="B" THEN M$(1,S)-
"D"
1140 IF S1=8 AND M$(8,S)="A" THEN M$(8,S)=
"C"
1150 SOUND 1,750,30
1160 R0=S1:CL=S:GOSUB 1190
1170 R0=F1:CL=F:GOSUB 1190
1180 RETURN
1190 N=R0:M=CL:C$=M$(N,M)
1200 GOSUB 810:RETURN
1210 REM >>>>>>> CONVERT <<<<<<<<
1220 REM
1230 F=ASC(LEFT$(F$,1))-64
1240 F1=9-VAL(RIGHT$(F$,1))
1250 S=ASC(LEFT$(S$,1))-64
1260 S1=9-VAL(RIGHT$(S$,1))
1270 RETURN
1280 R$=CHR$(M+64)+STR$(9-N)
1290 RETURN
1300 REM >>>>>>> VALIDATE <<<<<<<<
1310 REM
1320 GOSUB 1230:P$="B":P1$="D"

```

```

1330 IF PLAY=2 THEN P$="A":P1$="C"
1340 CHECK=1:C$=M$(F1,F):D$=M$(S1,S)
1350 IF C$="Ø" OR D$<>"Ø" THEN RETURN
1360 IF (C$="A" OR C$="C")AND PLAY=1 THEN
RETURN
1370 IF (C$="B" OR C$="D")AND PLAY=2 THEN
RETURN
1380 IF C$="A" AND S1<F1 THEN RETURN
1390 IF C$="B" AND S1>F1 THEN RETURN
1400 XD=S-F:YD=S1-F1
1410 IF SGN(XD)*(XD)=1 AND SGN(YD)*(YD)=1
THEN 1490
1420 IF (SGN(XD)*XD<>SGN(YD)*YD)OR (XD*XD<>
4) THEN RETURN
1430 X$=M$(F1+(YD/2),F+(XD/2))
1440 IF (X$="A" OR X$="C")ANDPLAY=2 THEN R
ETURN
1450 IF (X$="B" OR X$="D")ANDPLAY=1 THEN R
ETURN
1460 IF X$="Ø" THEN RETURN
1470 N=F1+(YD/2):M=F+(XD/2):M$(N,M)="Ø":C$=
"Ø":GOSUB 810
1480 GOSUB 1280:Z$(COUNT,1)=R$:Z$(COUNT,2)=
R$:COUNT=COUNT+1
1490 CHECK=Ø:RETURN
1500 REM >>>>>>>> TURN <<<<<<<<<
1510 REM
1520 CHAR ,25,20," :GOSUB 900
1530 GOSUB 1320
1540 IF CHECK=Ø THEN 1570
1550 COLOR 1,9,4:CHAR 1,25,20,"INVALID MOV
E"
1560 SOUND 1,600,25:SOUND 1,400,25:SOUND 1
,200,35:FOR Q=1TO200:NEXT:GOTO 1520
1570 GOSUB 1110
1580 Z$(COUNT,1)=F$:Z$(COUNT,2)=S$
1590 COUNT=COUNT+1:IF COUNT=101 THEN COUNT
=1
1600 RETURN
1610 REM >>>>> BOTH PLAYERS <<<<<<
1620 REM
1630 PLAY=1:GOSUB 1520:GOSUB 80
1640 PLAY=2:GOSUB 1520:GOSUB 80
1650 RETURN
1660 REM >>>>>>>> SAVE <<<<<<<<<

```

```
1670 REM
1680 OPEN 1,1,1,"DRAFTS"
1690 FOR N=1TO8:FOR M=1TO8
1700 PRINT#1,M$(N,M)
1710 NEXTM,N
1720 FOR N=1TO100
1730 PRINT#1,Z$(N,1):PRINT#1,Z$(N,2)
1740 NEXT N
1750 PRINT#1,COUNT:PRINT#1
1760 CLOSE 1
1770 SOUND 1,200,20:SOUND 1,400,20:SOUND 1
,600,30:RETURN
1780 REM >>>>>>>> LOAD <<<<<<<<
1790 REM
1800 SOUND1,200,20:SOUND1,400,20:SOUND1,60
0,30
1810 OPEN 1,1,0,"DRAFTS"
1820 FOR N=1TO8:FOR M=1TO8
1830 INPUT#1,M$(N,M)
1840 NEXTM,N
1850 FOR N=1TO100
1860 INPUT#1,Z$(N,1):INPUT#1,Z$(N,2)
1870 NEXT
1880 INPUT#1,COUNT
1890 CLOSE 1
1900 SOUND 1,200,20:SOUND 1,400,20:SOUND 1
,600,30:GOSUB 300:GOSUB 750
1910 RETURN
1920 REM >>>>>>>> REPLAY <<<<<<<<
1930 REM
1940 COUNT=0:PLAY=1:GOSUB 290:GOSUB 750
1950 COUNT=COUNT+1:IF Z$(COUNT,1)="FF" THE
N 2050
1960 F$=Z$(COUNT,1):S$=Z$(COUNT,2)
1970 GOSUB 1110:IF F$=S$ THEN 1950
1980 GETKEY A$:PLAY=2
1990 COUNT=COUNT+1:IF A$="#" OR Z$(COUNT,1)
)="FF" THEN 2050
2000 F$=Z$(COUNT,1):S$=Z$(COUNT,2)
2010 GOSUB 1110:IF F$=S$ THEN 1990
2020 GETKEY A$:PLAY=1
2030 IF A$="#" THEN 2050
2040 GOTO 1950
2050 IF PLAY=1 THEN 1630:ELSE 1640
```

8

Chess Duel



Computer Moderated Micro Chess

It will come as a surprise to many that before the age of chess computers and the microcomputer program, the game of chess was originally played by two human beings.

An intelligent chess playing program would take up an entire book on its own, so it was decided to develop a program which would be of genuine use to anyone who enjoyed playing chess or would like to learn more about it. Chess clubs could also find the program helpful.

The program can be used simply as a replacement for a chess board and pieces. The two players enter their moves in usual standard chess notation. However, extra features have been included to produce worthwhile advantages over the traditional board and pieces.

Only moves which are the correct colour, are from occupied squares and move to a square which can be reached by that piece, will be accepted. Pieces which are 'taken' in the process are removed from the board.

It is up to the humans to watch out for rule violations as in normal chess.

A useful feature is that the initial board may be set up for chess problems or chess variants and games of up to 100 moves can be stored and analysed step by step. Play can continue from any stage of a saved game or the present game can be re-started and then played from any subsequent move.

Summary of commands and options

1. Set up board. Enter 'Y' when asked. Then sequentially from the top left, place each piece using a single letter. Move on using 'Space'.

White pieces: P=pawn, R=rook, N=knight, K=king, Q=queen,
B=bishop.

Black pieces: the same but with 'Shift' held.

White always plays up the board.

2. Remove piece. ‘R’ then enter the position. Play continues with the next player.
 3. Save game. ‘S’.
 4. Load game. ‘L’, then press ‘Play’. This loads the first file found.
 5. Step through game. ‘Shift’ ‘3’ initiates step mode. ‘Space’ steps through each move. ‘Shift’ ‘3’ then ends mode.
 6. Quit. ‘Q’ resets for new game.


```

460 DATA 224,224,192,240,112,248,56,248
470 DATA 253,227,221,191,191,187,222,231
480 DATA 127,131,59,253,253,157,59,231
490 DATA 237,147,85,91,109,173,219,239
500 DATA 183,201,170,218,182,239,219,247
510 DATA 253,249,247,238,220,220,238,247
520 DATA 191,159,239,247,251,251,247,239
530 DATA 255,254,249,247,239,223,233,243
540 DATA 63,223,95,239,183,215,235,124
550 DATA 255,230,217,217,217,223,236,239
560 DATA 255,103,155,155,155,251,247,247
570 DATA 247,247,228,223,144,127,64,127
580 DATA 239,239,231,251,121,254,30,254
590 DATA 255,255,255,255,252,251,247,247
600 DATA 255,255,255,255,63,223,239,239
610 DATA 245,250,243,239,232,239,208,223
620 DATA 239,239,207,247,119,251,59,251
630 FOR N=1 TO 8:FOR M=1 TO 8:M$(N,M)="0":NEXTM,N
640 RESTORE 660:FOR N=1 TO 8:READ N$:M$(8,
N)=N$:M$(1,N)=CHR$(ASC(N$)+6)
650 M$(2,N)="G":M$(7,N)="A":NEXT:RETURN
660 DATA B,C,D,E,F,D,C,B
670 FOR N=1 TO 8:FOR M=1 TO 8
680 GOSUB 690:NEXTM,N:RETURN
690 C$=M$(N,M):IF C$>"F" THEN C$=CHR$(ASC(
C$)-6):R=1:ELSE R=0
700 A=N/2:B=M/2
710 IF (A=INT(A))AND(B=INT(B)) OR (A<>INT(
A))AND(B<>INT(B)) THEN SQ=0:ELSE SQ=1
720 IF C$="0"THEN 800
730 IF SQ=R THEN BAS=176:ELSE BAS=160
740 IF C$="A"THEN B$=CHR$(BAS+14)+CHR$(BAS
+15):ELSE B$=CHR$(BAS+10)+CHR$(BAS+11)
750 IF C$="A"THEN T$=CHR$(BAS+12)+CHR$(BAS
+13):GOTO 780
760 CH=2*(ASC(C$)-65):CH=10-CH
770 T$=CHR$(BAS+CH)+CHR$(BAS+CH+1)
780 IF R=1THENT$="({RVS ON}"+T$+"({RVS OFF})"
:B$="({RVS ON}"+B$+"({RVS OFF})"
790 COLOR 1,2,4:CHAR 1,M*2+1,N*2,T$:CHAR 1
,M*2+1,N*2+1,B$:RETURN
800 IF SQ=1THENT$=" " :ELSE T$="({RVS ON}
(RVS OFF)"
810 B$=T$:GOTO 790

```

```

820 COLOR1,2,4:FOR N=2TO15:CHAR1,25,N,"{RV
S ON}           (RVS OFF)":NEXT
830 COLOR1,3,2:FOR N=6TO13:CHAR1,26,N,""
":NEXT
840 COLOR1,16,4:CHAR1,26,3,"CHESS DUEL"
850 COLOR1,8,4:IF PLA=1THENNN$="WHITE":ELSE
N$="BLACK"
860 CHAR1,29,6,N$:CHAR1,27,8,"FROM:-":CHAR
1,27,11,"TO :-"
870 Y=8:GOSUB 890:F$=N$
880 Y=11:GOSUB 890:S$=N$:RETURN
890 N$="""
900 GETKEY A$:SOUND 1,500,10
910 IF A$="Q" THEN RUN
920 IF A$="S" THEN GOSUB 1350
930 IF A$="L" THEN GOSUB 1460:GOTO 820
940 IF A$="R" THEN GOTO 1690
950 IF A$="#AND COUNT>1 THEN Z$(COUNT,1)=
"FF":GOTO 1570
960 IF A$<"A"OR A$>"H" THEN GOTO 900
970 N$=A$:CHAR 1,34,Y,N$
980 GETKEYA$:SOUND1,600,10
990 IF A$<"1"OR A$>"9" THEN GOTO 980
1000 N$=N$+A$:CHAR1,34,Y,N$:RETURN
1010 GOSUB 1060:M$(S1,S)=M$(F1,F):M$(F1,F)
="0":SOUND 1,750,30
1020 R0=S1:CL=S:GOSUB 1040
1030 R0=F1:CL=F:GOSUB 1040:RETURN
1040 N=R0:M=CL:C$=M$(N,M)
1050 GOSUB 690:RETURN
1060 F=ASC(LEFT$(F$,1))-64
1070 F1=9-VAL(RIGHT$(F$,1))
1080 S=ASC(LEFT$(S$,1))-64
1090 S1=9-VAL(RIGHT$(S$,1)):RETURN
1100 R$=CHR$(M+64)+STR$(9-N):RETURN
1110 GOSUB 1060:CHECK=1:C$=M$(F1,F):D$=M$(
S1,S)
1120 IF C$="0" THEN RETURN
1130 IF (C$<"G" OR (D$>"F" AND D$<>"0")) AND
PLAY=2 THEN RETURN
1140 IF (C$>"F" OR (D$<"G" AND D$<>"0")) AND PLAY
=1 THEN RETURN
1150 IF C$="G" AND S1<F1 THEN RETURN
1160 IF C$="A" AND S1>F1 THEN RETURN
1170 IF D$="L" OR D$="F" THEN RETURN

```

```
1180 IF C$=="B"ORC$=="H"THEN 1940
1190 IF C$=="C"ORC$=="I"THEN 1950
1200 IF C$=="D"ORC$=="J"THEN 1990
1210 IF C$=="E"ORC$=="K"THEN 2020
1220 IF C$=="F"ORC$=="L"THEN 2060
1230 CHECK=0:IF C$<>"C"ANDC$<>"I"THEN GOTO
2070:ELSE RETURN
1240 W$="":FOR N=1TO12:W$=W$+" ":NEXT
1250 CHAR,25,20,W$:GOSUB 820:GOSUB 1110
1260 IF CHECK=0THEN1290
1270 COLOR 1,9,4:CHAR 1,25,20,"INVALID MOV
E"
1280 SOUND1,600,25:SOUND1,400,25:SOUND1,20
0,35:FOR Q=1TO200:NEXT:GOTO 1240
1290 GOSUB 1010
1300 Z$(COUNT,1)=F$:Z$(COUNT,2)=S$
1310 COUNT=COUNT+1:IF COUNT=101 THEN COUNT
=1
1320 RETURN
1330 PLAY=1:GOSUB 1240
1340 PLAY=2:GOSUB 1240:RETURN
1350 Z$(COUNT,1)="FF":SOUND 1,200,20:SOUND
1,400,20:SOUND 1,600,30
1360 OPEN1,1,1,"CHESS"
1370 FOR N=1TO8:FOR M=1TO8
1380 PRINT#1,M$(N,M):PRINT#1,X$(N,M)
1390 NEXTM,N
1400 FOR N=1TO100
1410 PRINT#1,Z$(N,1):PRINT#1,Z$(N,2)
1420 NEXT
1430 PRINT#1,COUNT:PRINT#1
1440 CLOSE1
1450 SOUND1,200,20:SOUND1,400,20:SOUND1,60
0,30:RETURN
1460 SOUND1,200,20:SOUND1,400,20:SOUND1,60
0,30
1470 OPEN 1,1,0,"CHESS"
1480 FOR N=1TO8:FOR M=1TO8
1490 INPUT#1,M$(N,M):INPUT#1,X$(N,M)
1500 NEXTM,N
1510 FOR N=1TO100
1520 INPUT#1,Z$(N,1):INPUT#1,Z$(N,2)
1530 NEXT
1540 INPUT#1,COUNT
1550 CLOSE1
```

```

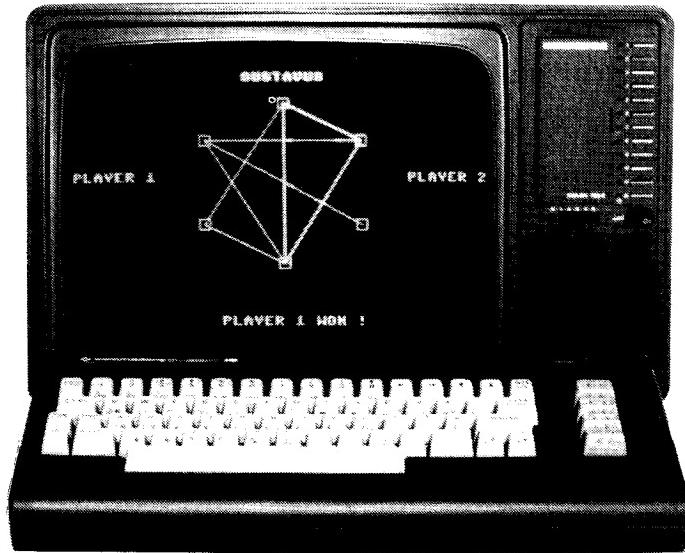
1560 SOUND1,200,20:SOUND1,400,20:SOUND1,60
0,30:GOSUB 70:GOTO 670
1570 COUNT=0:PLAY=1:GOSUB 60:FOR N=1TO8:FO
R M=1TO8:M$(N,M)=X$(N,M):NEXTM,N
1580 GOSUB 670
1590 COUNT=COUNT+1:IF Z$(COUNT,1)="FF" THE
N 1680
1600 F$=Z$(COUNT,1):S$=Z$(COUNT,2)
1610 GOSUB 1010
1620 GETKEYA$:PLAY=2
1630 COUNT=COUNT+1:IF A$="#" OR Z$(COUNT,1)
)="FF" THEN 1680
1640 F$=Z$(COUNT,1):S$=Z$(COUNT,2)
1650 GOSUB 1010
1660 GETKEYA$:PLAY=1
1670 IF A$="#" THEN 1680:ELSE 1590
1680 IF PLAY=1 THEN 1330:ELSE 1340
1690 SOUND1,200,20:SOUND1,400,20:SOUND1,60
0,30
1700 CHAR,30,22,"PIECE:--"
1710 GETKEYA$:SOUND1,400,10:IF A$<"A"OR A$>
"H"THEN 1710:ELSE CHAR,34,24,A$
1720 GETKEYB$:SOUND1,500,10:IF B$<"1"ORB$>
"8"THEN 1720:ELSE CHAR,35,24,B$
1730 F$=A$+B$
1740 S$=F$:GOSUB 1060:M$(F1,F)="0"
1750 CHAR,30,22,"      ":CHAR,34,24," "
1760 N=F1:M=F:GOSUB 690
1770 Z$(COUNT,1)=F$:Z$(COUNT,2)=S$:COUNT=C
OUNT+1:IF COUNT=101 THEN COUNT=1
1780 IF PLAY=1 THEN PLAY=2:GOSUB 1240
1790 GOTO 50
1800 FOR N=1TO8:FOR M=1TO8
1810 SOUND1,200,10:SOUND1,400,10
1820 T$="({RVS ON}{^U}{^I}{RVS OFF})":B$="({R
VS ON}{^J}{^K}{RVS OFF})"
1830 COLOR1,15,2:CHAR1,M*2+1,N*2,T$:CHAR1,
M*2+1,N*2+1,B$
1840 GETKEYA$:IF A$=CHR$(13) OR A$=" " THEN C$=
"0":GOTO 1890
1850 RESTORE1920:FLAG=0
1860 FOR Q=1TO12:READA,B$
1870 IF ASC(A$)=ATHENC$=B$:FLAG=1
1880 NEXT:IF FLAG=0 THEN 1840
1890 M$(N,M)=C$

```

```
1900 GOSUB 690:NEXTM,N
1910 RETURN
1920 DATA 80,A,82,B,78,C,66,D,81,E,75,F
1930 DATA 208,G,210,H,206,I,194,J,209,K,20
3,L
1940 IF F1<>S1ANDF<>S THENRETURN:ELSE GOTO
1230
1950 TA=(F1-S1)*SGN(F1-S1):TB=(F-S)*SGN(F-
S)
1960 IF TA=2ANDTB=1THEN 1230
1970 IF TA=1ANDTB=2THEN 1230
1980 RETURN
1990 XD=F-S:XD=SGN(XD)*XD
2000 YD=F1-S1:YD=SGN(YD)*YD
2010 IF XD<>YDTHEN RETURN:ELSE GOTO 1230
2020 XD=F-S:XD=SGN(XD)*XD
2030 YD=F1-S1:YD=SGN(YD)*YD
2040 IF XD=YDTHEN 1230
2050 IF F1<>S1ANDF<>S THEN RETURN
2060 IF SGN(F1-S1)*(F1-S1)<>1 AND SGN(F-S)
*(F-S)<>1 THENRETURN:ELSE GOTO 1230
2070 ND=0:MD=0
2080 IF F1>S1THEN ND=-1
2090 IF S1>F1THEN ND=1
2100 IF F>STHEN MD=-1
2110 IF S>FTHEN MD=1
2120 CN=F1:CM=F
2130 CN=CN+ND:CM=CM+MD:IF CN=S1ANDCM=S THE
N RETURN
2140 IF M$(CN,CM)<>"0"THENCHECK=1:RETURN
2150 GOTO 2130
```

9

Gustavus



A Novel Strategy Game for Two

As far as I am aware we are the first to produce a computer implementation of *Simple Simmons*, a game usually played with pencil and paper.

Simple Simmons or *Sim* was invented by a chap called Gustavus Simmons. We have decided to call this computer version of the game *Gustavus* as it sounds more grand than its paper cousin.

The game requires players alternately to draw a line between two points. The first to form a triangle of his own colour, whose points each rest on one of the original points, is the loser. The trick is to force your opponent into making the fatal triangle first.

The computer will take care of the turns together with the colour changes and will indicate the winner when a triangle is formed.

Lines are drawn by moving clockwise or anti-clockwise, using either left or right cursor keys. When on the first point from which a line is required press the space bar. A bleep signifies acceptance of the line origin. The cursor is next moved to the destination location, then as the space key is pressed the line will be drawn. If an invalid line is

attempted a sound will indicate the fact. New start and finish points will then be required.

The game cannot end in a draw and there is no apparent advantage gained by being the first player. This game is very good for developing spacial awareness!

```

10 REM >>>>GUSTAVUS<<<<<
20 REM >>>>DAVIS/ISSI/JIM<<
30 COLOR0,1:COLOR4,1:VOL8:GOSUB120
40 GOSUB80:GOSUB140:GOSUB330:FORN=0TO999:N
EXT:IFPL=1THENWI=2:ELSEWI=1
50 IFWI=1THENC=12:ELSEC=14
60 COLOR1,C,4:A$="PLAYER"+STR$(WIN)+" WON
!":CHAR1,14,24,A$
70 GETA$:IFA$=""THEN70:ELSE40
80 GRAPHIC1,1:COLOR1,2,7:CHAR1,16,0,"GUSTA
VUS":COLOR1,8,6
90 FORN=1TO6:LOCATEP(N,1)+4,P(N,2)-4:DRAWT
0+0,+8
100 DRAWTO-8,+0:DRAWTO+0,-8:DRAWTO+8,+0:LO
CATEP(N,1),P(N,2):NEXT:COLOR1,12,4
110 CHAR1,0,10,"PLAYER 1":COLOR1,14,4:CHAR
1,32,10,"PLAYER 2":RETURN
120 DIMP(6,2),D(6,6):RESTORE130:FORN=1TO6:
READP(N,1):READP(N,2):NEXT:RETURN
130 DATA160,150,220,120,220,55,160,25,100,
55,100,120
140 FORN=1TO6:FORM=1TO6:D(N,M)=0:NEXT:NEXT
:RETURN
150 LE=0:RI=0:FI=0:GETA$:IFA$="(CUR L)"THE
NLE=1:ELSEIFA$="(CUR RT)"THENRI=1
160 IFA$=""THENFI=1
170 J=JOY(1)ORJOY(2):IFJ=3THENRI=1:ELSEIFJ
=7THENLE=1
180 IFJ=128THENFI=1:RETURN:ELSERETURN
190 PN=1:IFPL=1THENCOLOR1,12,4:ELSECOLOR1,
14,4
200 LOCATEP(PN,1)-5,P(PN,2)+5:GOSUB380:GOS
UB150:IFFI=1THEN250
210 LOCATEP(PN,1)-5,P(PN,2)+5:GOSUB380:IFL
E=1THENPN=PN-1
220 IFRI=1THENPN=PN+1
230 IFPN=7THENPN=1:ELSEIFPN=0THENPN=6
240 GOTO200
250 SOUND1,800,10:RETURN

```

260 LOCATEP(FR,1)-5,P(FR,2)+5:GOSUB380:LOC
ATEP(DE,1)-5,P(DE,2)+5:GOSUB380
270 RETURN
280 GOSUB190:FR=PN:GOSUB190:DE=PN:GOSUB260
290 IFFR=DEORD(FR,DE)<>0THEN SOUND3,800,5:S
OUND3,1010,5:GOTO280
300 LOCATEP(FR,1),P(FR,2):DRAWTOP(DE,1),P(
DE,2):D(DE,FR)=PL:D(FR,DE)=PL
310 FOR N=1 TO 6: IF D(DE,N)=PLANDD(N,FR)=PLTHE
NDD=1:GOSUB350
320 NEXT:RETURN
330 DD=0:PL=1:GOSUB280: IF DD=1 THEN RETURN:EL
SE PL=2:GOSUB280
340 IF DD=1 THEN RETURN:ELSE 330
350 COLOR1,2,7:LOCATEP(FR,1),P(FR,2)
360 DRAWTOP(N,1),P(N,2):DRAWTOP(DE,1),P(DE
,2):DRAWTOP(FR,1),P(FR,2)
370 SOUND3,999,20:RETURN
380 IF PL=1 THEN GOSUB430:ELSE GOSUB440
390 IF PL=1 THEN NC=12:ELSE NC=14
400 COLOR1,C,6:LOCATE+9,-9:DRAWTO+0,+8:DRA
WTO-8,+0:DRAWTO+0,-8:DRAWTO+8,+0
410 IF PL=1 THEN NC=12:ELSE NC=14
420 COLOR1,C,4:DRAWTO+0,+8:DRAWTO-8,+0:DRA
WTO+0,-8:DRAWTO+8,+0:LOCATE-9,+9:RETURN
430 COLOR1,2,5:CHAR1,0,11,"{^*}{^*}{^*}{^*}
{^*}{^*}{^*}":COLOR1,1:CHAR1,0,11,"
":RETURN
440 COLOR1,2,5:CHAR1,32,11,"{^*}{^*}{^*}{^*}{^*}
{^*}":CHAR0,32,11,"":RE
TURN

10

Dice Derby



Shake!

This is the first of a dice trio to be followed by *Snake Eyes* and *Craps*. All three use the same central routines and a menu is provided to select between the games. This will produce an enjoyable set of dice games and save a lot of typing.

Dice Derby is a race game in which the dice total is used to determine how far each horse travels. It can be played by two to four players and has an attractive shaking cup feature.

The fire button is pressed to start the roll and again to finish. This means that each player can exercise control over the duration of each shake.

The characters will race along until one is declared the winner. The winner is not determined until the last throw of the round and so some very close 'photo finishes' are possible.

The listing for all three games follows the *Craps* instructions.

11

Snake Eyes



Rattle!

This is the second of the three dice games. The instructions for throwing the dice are the same for all the games.

The game is for two to four players and includes on-screen betting. At the start the target pot is entered, which may be between £40 and £400. The bank starts at £0.

Each player then rolls the dice. If a double is thrown then the player wins the bank amount. If 'Snake Eyes' (two ones) is thrown then the player gets twice the bank total. The bank resets to zero when it pays out.

After each round the bank gets £5 if the target set is below £200 and £10 if the target is above £200.

The game continues until a player's cash reaches the target score.

It may be interesting to change the amount to the bank after each round to provide a larger 'pot'!

The listing follows the *Craps* instructions.

12 Craps



... and Roll!

Here it is, the micro version of the American dice game featured in gangster movies and TV shows.

Two to four players can now pretend that they are in Las Vegas courtesy of the Commodore 16.

At the start of the game each player has £100 (you can change it to dollars if you want more realism!). A bet is placed up to the total holding of each player on his turn. Then the fun begins.

If a 7 or 11 is thrown first then they are paid evens. If 2, 3 or 12 is thrown the turn ends and the stake is lost. Any other total becomes the *point*. This is then the total to roll for.

The player continues to roll until the point is reached, paying evens, or a 7 is thrown in which case it is the next player's turn.

If a player loses all his money then he drops out and the game continues until only one player is left.

The listing which follows will produce all three dice programs with menu selection.

```
10 REM >>>> DICERAMA * ISSI/GREG/JIM<<<
20 POKE52,55:POKE54,55:POKE56,55:GOSUB2160
:SYS882:VOL8:POKE214,0
30 POKE65298,192:POKE65299,(PEEK(65299)AND
31 OR56
40 XX=RND(-TI)
50 GOSUB210:GOSUB910
60 SCNCLR:GOSUB620
70 COLOR1,6,3:CHAR1,13,2,"1 > DICE DERBY"
80 CHAR1,13,4,"2 > SNAKE EYES":CHAR1,13,6,
"3 > CRAPS"
90 GETKEYA$:SOUND1,500,12
100 ONVAL (A$)GOTO1060,1380,1670
110 GOTO90
120 COLOR1,3,3:CHAR1,13,8,"PRESS SPACE BAR
"
130 GETKEYA$
140 IF A$<>" "THEN130
150 GOTO60
160 POKE210,150:POKE211,0:POKE212,7:SYS828
170 POKE210,134:POKE211,113:POKE212,12:SYS
828:RETURN
180 POKE210,134:POKE211,0:POKE212,12:SYS82
8
190 POKE210,150:POKE211,113:POKE212,7:SYS8
28:RETURN
200 REM >>>>> DEFINE CHARACTERS <<<<<
210 RESTORE230:FORT=0T079:READA:POKE14616+
T,A:NEXT
220 FORT=0T0415:READA:POKE14848+T,A:NEXT:R
ETURN
230 DATA0,0,0,24,24,0,0,0,3,3,0,0,0,0,192,
192
240 DATA3,3,0,24,24,0,192,192,195,195,0,0,
0,0,195,195
250 DATA195,195,0,24,24,0,195,195,219,219,
0,0,0,0,219,219
260 DATA60,66,129,129,129,129,66,255,24,24
,24,24,24,24,24
270 DATA255,66,129,129,129,129,66,60,18,34
,53,120,248,136,136,136
280 DATA0,0,0,0,0,1,3,2
290 DATA0,0,0,127,128,0,255,0
```

```
300 DATA0,0,0,248,24,41,203,74
310 DATA0,0,0,248,24,40,200,72
320 DATA2,2,2,2,2,2,2,2
330 DATA74,74,74,74,74,74,74,82
340 DATA72,72,72,72,72,72,72,80
350 DATA2,3,0,0,0,0,0,0
360 DATA0,255,0,0,0,0,0,0
370 DATA98,195,0,0,0,0,0,0
380 DATA96,192,0,0,0,0,0,0
390 DATA1,2,2,4,4,8,16,16,240,143,64,64,32
,32,16,16
400 DATA0,0,240,15,0,0,0,0,0,0,0,0,224,31,
0,0
410 DATA0,0,0,0,0,0,248,7,0,0,0,0,0,0,0,22
4
420 DATA32,32,32,64,64,64,128,128,8,8,4,4,
4,2,2,2
430 DATA31,0,0,0,0,0,0,0,0,240,8,8,8,4,4,4
440 DATA128,128,128,128,128,128,128,128,64
,64,64,32,32,32,16
450 DATA2,4,4,4,4,8,8,16,4,4,8,8,16,240,0,
0
460 DATA16,16,8,8,4,4,2,1,16,32,32,64,64,1
95,156,224
470 DATA0,0,0,15,112,128,0,0,0,31,224,0,0,
0,0,0
480 DATA255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,15,2
40
490 REM * DICE UPRIGHT *
500 DATA0,0,0,3,12,48,64,128,0,15,112,128,
0,0,0,0
510 DATA192,56,7,0,0,0,0,0,0,0,0,192,32,24
,6,1
520 DATA224,152,134,65,64,64,32,32,0,0,0,1
28,96,30,1,0
530 DATA0,0,0,0,0,255,0,0,0,0,0,3,28,224
,0
540 DATA3,13,49,193,2,2,2,2,32,16,16,16,8,
8,8,8
550 DATA4,4,4,4,8,8,8,8,4,4,4,4,2,2,2,2,1
560 DATA16,16,16,32,32,32,32,32,1,1,1,1,0,
0,0,0
570 DATA0,0,0,0,128,128,128,128,64,64,64,6
4,64,128,128,128
580 DATA128,128,128,64,64,64,64,128,128
```

```

,128,0,0,0,0,0
590 DATA0,0,0,1,1,1,1,32,32,32,32,56,14,
3,0
600 DATA2,2,2,2,14,112,128,0
610 REM >>>>>>> SCREEN <<<<<<<<
620 COLOR0,1:COLOR4,1:SCNCLR:COLOR1,2,7
630 D1$=CHR$(96)+CHR$(97)+CHR$(98)+CHR$(97)
)+CHR$(99)
640 D2$=CHR$(100)+CHR$(32)+CHR$(101)+CHR$(32)
+CHR$(102)
650 D3$=CHR$(103)+CHR$(104)+CHR$(105)+CHR$(104)
+CHR$(106)
660 L1$="" :FORT=107T0112:L1$=L1$+CHR$(T):N
EXT
670 L2$=CHR$(113)+CHR$(114)+"      "+CHR$(115)
+CHR$(116)
680 L3$=CHR$(117)+CHR$(100)+"      "+CHR$(10
0)
690 L4$=CHR$(118)+CHR$(119)+"      "+CHR$(126)
+CHR$(120)
700 L5$="" :FORT=121T0125:L5$=L5$+CHR$(T):N
EXT
710 U5$=CHR$(172)+CHR$(173)+"      "+CHR$(174)
720 U6$=CHR$(32)+CHR$(175)+CHR$(32)+CHR$(1
77)+CHR$(176)
730 U7$=CHR$(32)+CHR$(178)+CHR$(165)+CHR$(1
79)+CHR$(32)
740 FORX=0T038:CHAR1,X,13,"--":NEXT
750 COLOR1,1:CHAR1,6,20,D1$:CHAR1,6,21,D2$:
CHAR1,6,22,D3$
760 CHAR1,12,17,L1$:CHAR1,12,18,L2$:CHAR1,
12,19,L3$:CHAR1,12,20,L4$
770 CHAR1,12,21,L5$:COLOR1,2,7
780 SYS913:GOSUB180
790 RETURN
800 REM >>>>>>> RATTLE <<<<<<<<<
810 GOSUB180
820 SYS913
830 SOUND2,600,5:SOUND2,0,0
840 D1=INT(RND(1)*6)+1
850 D2=INT(RND(1)*6)+1
860 GETA$:IFA$<>" "THEN820
870 SOUND1,400,25
880 CH=D1+34:POKE3919,CH:CH=D2+34:POKE3921
,CH

```

```

890 RETURN
900 REM >>>>>> INITIALIZE <<<<<<<
910 DIMS(7),P(4,3),O(4):RETURN
920 REM >>>>>>> CLEAR <<<<<<<<<
930 CL=3112:FORT=1TO12:FORN=0TO39:POKECL+N
,32:NEXT:CL=CL+40:NEXT
940 RETURN
950 REM >>>>>>> PLAYERS <<<<<<<<
960 GOSUB930:COLOR1,6,4:CHAR1,13,5,"PLAYER
S ? 2 - 4"
970 GETKEYA$
980 IFA$<"2"ORA$>"4"THEN970
990 SOUND1,350,25:SOUND1,200,25:SOUND1,100
,25
1000 GOSUB930
1010 FORN=1TO4:P(N,1)=0:NEXT
1020 PY=VAL(A$)
1030 FORN=1TOPY:O(N)=1:NEXT
1040 RETURN
1050 REM >>>>>> DICE DERBY <<<<<<<<
1060 GOSUB960
1070 FORN=1TO4
1080 P(N,1)=0:P(N,2)=3*N-2
1090 NEXT:COLOR1,5,3:RESTORE1110
1100 FORY=1TO12:READA:CHAR1,38,Y,CHR$(A):N
EXT
1110 DATA41,42,42,70,73,78,73,83,72,42,42,
43
1120 FORW=1TOPY:GOSUB1360:NEXT
1130 FORW=1TOPY:COLOR1,W+3,4
1140 PL$="PLAYER "+STR$(W):CHAR1,22,14,PL$
1150 SOUND1,300,40
1160 GETKEYA$
1170 IFA$<>" "THEN1160
1180 GOSUB810
1190 GOSUB160
1200 FORD=1TO(D1+D2)
1210 SOUND2,400,1:IFP(W,1)>37THEN1220:ELSE
GOSUB1350
1220 P(W,1)=P(W,1)+1:IFP(W,1)>37THEN1240
1230 SOUND2,200,1:GOSUB1360
1240 NEXTD:NEXTW
1250 HO=0:HI=0:WI=0:FORN=1TO4
1260 IFP(N,1)>37THENHO=P(N,1)
1270 IFHO>HITHENHI=HO:WI=N

```

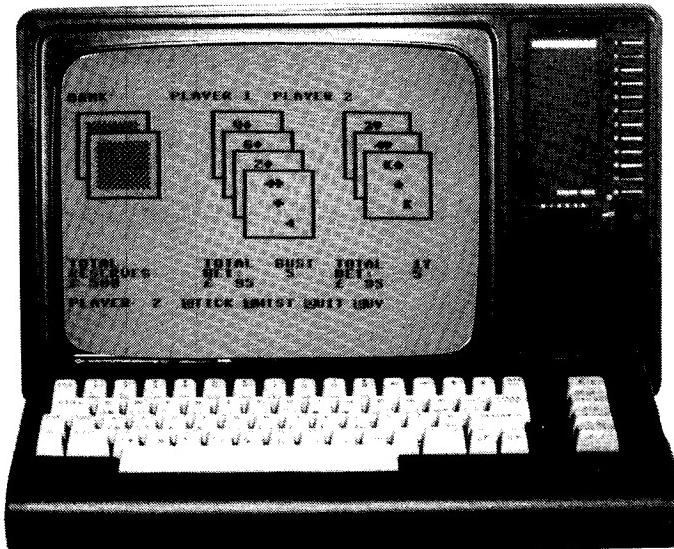
```
1280 NEXT
1290 IFWI=0THEN1130
1300 GOSUB930
1310 COLOR1,4,6:PL$="PLAYER "+STR$(WI)+" W
ON !"
1320 CHAR1,13,5,PL$
1330 FORT=1T06:FORS=800T0500STEP-20:SOUND1
,S,2:NEXT:NEXT
1340 GOTO120
1350 CHAR1,P(W,1),P(W,2)," ":RETURN
1360 COLOR1,W+3,4:CHAR1,P(W,1),P(W,2),CHR$(
44):RETURN
1370 REM >>>>>>> SNAKE EYES <<<<<<
1380 GOSUB960
1390 COLOR1,7,4:CHAR1,13,5,"ENTER WIN SCOR
E ?"
1400 CHAR1,28,5," ":INPUTLM
1410 IFLM>4000RLM<40THEN1400
1420 UA=5:GOSUB930:IFLM>200THENUA=10
1430 COLOR1,8,5:CHAR1,5,0,"1":CHAR1,15,0,"
2":CHAR1,25,0,"3":CHAR1,35,0,"4"
1440 COLOR1,4,6:FORT=1T07STEP6:FORN=0T038:
CHAR1,N,T,"-":NEXT:NEXT
1450 KT=0
1460 KT=KT+UA
1470 PR=1
1480 COLOR1,8,6:FORN=1T04:CH$=STR$(P(N,1))
:CHAR1,N*10-6,3,CH$:NEXT
1490 COLOR1,PR+3,4:PL$="PLAYER "+STR$(PR):
CHAR1,22,14,PL$
1500 GOSUB1530:IFP(PR,1)>LMTHEN1650
1510 PR=PR+1:IFPR>PYTHEN1460
1520 GOTO1480
1530 GETKEYA$
1540 COLOR1,9,6:BA$="BANK STANDS AT "+STR$(
KT)+" ":"CHAR1,10,12,BA$"
1550 GOSUB810:GOSUB160
1560 IFD1=1ANDD2=1THENGOSUB1590
1570 IFD1=D2ANDD1<>1THENGOSUB1620
1580 RETURN
1590 SOUND1,400,25:SOUND1,300,25
1600 SOUND1,200,25:SOUND1,300,25
1610 P(PR,1)=P(PR,1)+KT*2:KT=0:RETURN
1620 SOUND2,600,25:SOUND2,400,50
1630 SOUND2,600,25:SOUND2,200,50
```

```
1640 P(PR,1)=P(PR,1)+KT:KT=0:RETURN
1650 WI=PR:GOTO1300
1660 REM >>>>>>> CRAPS <<<<<<<<
1670 GOSUB960
1680 COLOR1,8,5:CHAR1,5,0,"1":CHAR1,15,0,"2":CHAR1,25,0,"3":CHAR1,35,0,"4"
1690 COLOR1,4,6:FORT=1TO7STEP6:FORN=0TO38:CHAR1,N,T,"-":NEXT:NEXT
1700 COLOR1,5,6
1710 FORN=1TO4
1720 P(N,1)=100:NEXT
1730 FORW=1TOPY:FORN=1TO4
1740 COLOR1,8,6:CH$=STR$(P(N,1))+""":CHAR1,N*10-7,3,CH$":NEXT
1750 IFP(W,1)<1THEN1920
1760 BT$="PLAYER "+STR$(W)+" BET":COLOR1,6,5:CHAR1,10,11,BT$
1770 CHAR1,24,11," ":"INPUTBT
1780 IFBT<1ORBT>P(W,1)THENSOUND1,300,15:GOT01770
1790 GOSUB810:GOSUB160
1800 TT=D1+D2
1810 IFTT=70RTT=11THEN2060
1820 IFTT=20RTT=30RTT=12THEN2110
1830 PT=TT:COLOR1,9,5:CHAR1,10,12,"ROLL AGAIN . . ."
1840 GETKEYA$"
1850 IFA$(<)" "THEN1840
1860 CHAR1,10,12,"SPACES" :REM 15
1870 GOSUB810:GOSUB160
1880 TT=D1+D2
1890 IFTT=PTTHEN2060
1900 IFTT=7THEN2110
1910 GOT01830
1920 NEXTW
1930 FG=0:CN=0
1940 FORN=1TOPY
1950 IFCN=1THEN1980
1960 IFP(N,1)>0ANDFG=1THENCN=1
1970 IFP(N,1)>0THENFG=1
1980 NEXTN
1990 IFCN=1THEN1730
2000 WI=0
2010 FORN=1TOPY
```

```
2020 IFWI<>0THEN2040
2030 IFF(N,1)>0THENWI=N
2040 NEXTN
2050 GOTO1300
2060 FORT=11TO12:FORN=0TO30:CHAR1,N,T," ":""
NEXT:NEXT
2070 SOUND1,300,25:SOUND2,200,25:SOUND2,10
0,40
2080 COLOR1,3,4:CHAR1,10,11,"YOU WIN !"
2090 FORT=1TO550:NEXT
2100 P(W,1)=P(W,1)+BT:GOTO1920
2110 FORT=11TO12:FORN=0TO30:CHAR1,N,T," ":""
NEXT:NEXT
2120 SOUND1,100,25:SOUND1,200,25:SOUND1,40
0,40
2130 COLOR1,3,4:CHAR1,10,11,"YOU LOSE !":F
ORT=1TO550:NEXT
2140 P(W,1)=P(W,1)-BT:GOTO1920
2150 POKE65298,196:POKE65299,209:STOP
2160 RESTORE2190:FORN=0TO30:READA:POKE882+
N,A:NEXT
2170 FORN=0TO42:READA:POKE828+N,A:NEXT
2180 FORN=0TO85:READA:POKE913+N,A:NEXT:RET
URN
2190 DATA160,0,132,212,132,214,169,208,133
,213,169,56,133,215,162,3
2200 DATA177,212,145,214,200,208,249,230,2
13,230,215,202,16,242,96
2210 DATA169,10,133,209,165,210,133,208,16
2,6,164,212,165,211,145,208
2220 DATA136,16,251,202,48,5,32,88,3,176,2
39,96,24,165,208,105,40,133,208,165
2230 DATA209,105,0,133,209,56,96
2240 DATA165,214,73,1,133,214,24,105,149,1
33,208,169,14,133,209,162
2250 DATA48,160,6,189,182,3,145,208,202,13
6,16,247,138,48,5,32,88,3,176,237,96
2260 DATA32,32,114,101,115,32,32,32,32,111
,32,113,112,32
2270 DATA32,108,109,32,32,110,32,32,106,32
,32,32,107,32
2280 DATA32,104,32,32,32,105,32,32,99,100,
101,102,103,32
2290 DATA32,95,96,93,97,98,32
```

13

Pontoon



A Pair can Break the Bank

In this computer version of the well known card game the bank starts with £500 and each player has £100. One or two players each play against the bank and aim to get a total of 21, which is Pontoon.

One card is first dealt face up to each player and one face down to the bank. A bet up to the total cash available may then be placed. The next card is dealt and various options are given to each player on their turn.

'S'tick. If the total is 16 or over then pressing 'S' will pass play on.

'T'wist. If another card is required this option may be taken by pressing 'T'.

'B'uy. Used when a hand looks good, to double the stake.

An ace will be taken to have the best value, i.e. 1 or 11 dependent upon the other cards. If at any time the card total exceeds 21 then play passes on. When both players have finished, the bank will then play its own hand to determine payout. It should be remembered that play is between the players and the bank, not against each other.

Wins and losses are calculated as follows: highest score is Pontoon with two cards, e.g. an Ace and a Queen. The next highest score is a 5-card trick, which may be any five cards which do not exceed 21 in total. Pontoon with any card permutation is the third highest score. If none of the above exist then the highest score wins. Where the bank's total equals that of a player then the bank wins.

The game continues until each player has lost all available cash or until the bank is bust.

```

10 REM ** PONTOON * ANDY/JIM/GREG **
20 GOSUB2730:POKE52,55:POKE54,55:POKE56,55
:GOSUB2750:SYS882:VOL8
30 DIMPA(52):DIMAV(52):DIMPA$(5,13):DIMCC(
3)
40 DIMCX(4),CY(4),PC(3,5),BC(5),NO(4),PL(2)
),MO(2),P$(2)
50 GOSUB2400:A=65298:POKEA,PEEK(A)AND251:P
OKEA+1,(PEEK(A+1)AND3)OR56
60 XX=RND(-TI):GOSUB1600
70 MO(1)=100:MO(2)=100:MC=500
80 CC(1)=20:CC(2)=33:CC(3)=6:PL(1)=1:PL(2)
=1:GOSUB1120
90 GOSUB1230
100 SCNCLR:IFNP=1THENPL(2)=0
110 IFPL(1)=0ANDPL(2)=0THEN2630
120 IFPL(1)=0THENBE(1)=0
130 IFPL(2)=0THENBE(2)=0
140 GOSUB1660
150 GOSUB1660:CX(1)=15:CX(2)=28:CY(1)=4:CY
(2)=4:BX=2:BY=4
160 GOSUB1810
170 FORQ=1TONP
180 IFPL(Q)=0THEN210
190 GOSUB1930
200 GOSUB2450
210 NEXTQ
220 CX=BX:CY=BY:FORQ=1TONC
230 GOSUB1520:CY=CY+2:CX=CX+1:NEXTQ
240 GOSUB2200
250 FORQ=1TONP:IFPL(Q)=0THEN280
260 CY=CY(Q):CX=CX(Q)
270 CY=CY+2:CX=CX+1:GOSUB2340:GOSUB2450
280 NEXTQ
290 CX=BX+1:CY=BY+2:GOSUB1480

```

```

300 GOSUB1520:GOSUB2010
310 REM ** SCORE LOGIC **
320 Q=3:IFNP=2THEN340
330 IFSC(1)>21THENP$(1)="L":GOT0500
340 IFSC(1)>21ANDSC(2)>21THENP$(1)="L":P$(2)="L":GOT0500
350 FORW=1TONC
360 CX=BX:CY=(BY-2)+W*2
370 PC(3,W)=BC(W)
380 GOSUB1760:GOSUB1310
390 BX=BX+1
400 NEXTW
410 BX=CX:BY=CY
420 FORW=1TONC
430 PC(3,W)=BC(W):NEXT
440 Q=3:NO(3)=NC:GOSUB2450
450 IFSC(3)<16THENGOSUB1480:ELSEGOT0500
460 BX=BX+1:BY=BY+2:PC(3,NC)=BC(NC):Q=3:CX
=BX:CY=BY:W=NC:GOSUB1760:GOSUB1310
470 NO(3)=NC:GOSUB2450
480 IFNC=5THEN500:ELSEGOT0450
490 F=INT(RND(1)*2)+17:IFSC(3)<FTHENGOT046
500
500 Q=1
510 IFPL(Q)=0THEN640
520 IFSC(Q)>21THENP$(Q)="L":GOT0640
530 IFSC(3)>21THENP$(Q)="W":GOT0640
540 IFNC=5ANDNO(Q)=5THENP$(Q)="L":GOT0640
550 IFNO(Q)=5THENP$(Q)="W":GOT0640
560 IFNC=5THENP$(Q)="L":GOT0640
570 IFSC(3)>SC(Q)THENP$(Q)="L":GOT0640
580 IFSC(3)=21ANDNC=2THENP$(Q)="L":GOT0640
590 IFNO(Q)=5ANDNC<5THENP$(Q)="W":GOT0640
600 IFSC(Q)=21ANDNO(Q)=2THENP$(Q)="W":GOT0
640
610 IFSC(Q)>SC(3)THENP$(Q)="W":GOT0640
620 IFSC(Q)=SC(3)THENP$(Q)="L":GOT0640
630 IFSC(3)>SC(Q)THENP$(Q)="L":GOT0640
640 IFNP=2ANDQ=1THENQ=2:GOT0510
650 IFNP=1THEN700
660 IFP$(1)="L"ANDP$(2)="L"THEN860
670 IFP$(1)="W"ANDP$(2)="W"THEN820
680 IFP$(1)="W"ANDP$(2)="L"THEN930
690 IFP$(1)="L"ANDP$(2)="W"THEN980
700 IFP$(1)="W"THEN740

```

```

710 IFP$(1)="L"THEN780
720 END
730 REM ** PLAYER WIN **
740 MC=MC-BE(1):MO(1)=MO(1)+BE(1)*2:GOSUB2
670:IFMC<1THEN1030
750 GOSUB2550
760 GOSUB2550:X=0:Y=23:GOSUB1080:PRINT"(RE
D)PLAYER WINS"::GOSUB2560:GOTO90
770 REM ** BANK WIN **
780 MC=MC+BE(1):GOSUB2550:GOSUB2700
790 X=0:Y=23:GOSUB1080:PRINT"(RED)BANK WIN
S"::GOSUB2560:IFMO(1)<1THENPL(1)=0
800 GOTO90
810 REM ** BOTH WIN **
820 MO(1)=MO(1)+BE(1)*2:MO(2)=MO(2)+BE(2)*
2:GOSUB2670:MC=MC-BE(1)-BE(2)
830 IFMC<1THEN1030
840 GOSUB2550:GOTO90
850 REM ** BOTH LOSE **
860 MC=MC+BE(1)+BE(2):GOSUB2550:GOSUB2700:
IFMO(1)<1THENPL(1)=0
870 IFMO(2)<1THENPL(2)=0
880 IFPL(1)=0ANDPL(2)=1THENME$="(RED)PLAYE
R 2 LOSES"
890 IFPL(1)=1ANDPL(2)=0THENME$="(RED)PLAYE
R 1 LOSES"
900 IFPL(1)=1ANDPL(2)=1THENME$="(RED)BOTH
PLAYERS LOSE"
910 X=0:Y=23:GOSUB1080:PRINTME$::GOSUB2560
:GOTO90
920 REM ** PLAYER 1 WIN **
930 MC=MC+BE(2):MO(1)=MO(1)+BE(1)*2:MC=MC-
BE(1):GOSUB2670
940 IFMO(2)<1THENPL(2)=0
950 IFMC<1THEN1030
960 GOSUB2550:X=0:Y=23:GOSUB1080:PRINT"(RE
D)PLAYER 1 WINS"::GOSUB2560:GOTO90
970 REM ** PLAYER 2 WIN **
980 MC=MC+BE(1):MO(2)=MO(2)+BE(2)*2:MC=MC-
BE(2):GOSUB2670
990 IFMO(1)<1THENPL(1)=0
1000 IFMC<1THEN1030
1010 GOSUB2550:X=0:Y=23:GOSUB1080:PRINT"(R
ED)PLAYER 2 WINS"::GOSUB2560:GOTO90
1020 REM ** BANK BROKE **

```

```

1030 GOSUB2550:X=0:Y=23:GOSUB1080
1040 PRINT"(RED)WELL DONE, YOU BROKE THE BA
NK";:GOSUB2670:GOSUB2670:GOSUB2560
1050 SCNCLR:PRINT"PLAYER 1:"MO(1);
1060 IFNP=2THENPRINT"PLAYER 2:"MO(2)
1070 GOSUB2560:GOSUB2560:GOT060
1080 POKE208,X:POKE209,Y:SYS828:RETURN
1090 FORN=1TO4:FORT=1TO4:PRINTPA$(N,T):NEXT
T:NEXT
1100 STOP
1110 REM ** GENERATE PACK **
1120 FORSU=1TO4
1130 FORCA=1TO13
1140 PA$(SU,CA)=CHR$(SU+96)
1150 IFCA<11ANDCA>1THENPA$(SU,CA)=PA$(SU,C
A)+CHR$(32)+CHR$(CA+48)
1160 IFCA=1THENPA$(SU,CA)=PA$(SU,CA)+" A"
1170 IFCA=11THENPA$(SU,CA)=PA$(SU,CA)+" J"
1180 IFCA=10THENPA$(SU,CA)=PA$(SU,CA)+" 10"
1190 IFCA=12THENPA$(SU,CA)=PA$(SU,CA)+" Q"
1200 IFCA=13THENPA$(SU,CA)=PA$(SU,CA)+" K"
1210 NEXT:NEXT:RETURN
1220 REM ** SHUFFLE **
1230 FORN=1TO52:PA(N)=N:AV(N)=1:NEXT
1240 FORA=1TO30
1250 N1=INT(RND(1)*51)+1:N2=INT(RND(1)*51)
+1
1260 CC=PA(N1)
1270 PA(N1)=PA(N2)
1280 PA(N2)=CC
1290 NEXT:RETURN
1300 REM ** DRAW CARD **
1310 X=CX-1:Y=CY:GOSUB1080
1320 COLOR1,1:PRINTCHR$(111);
1330 PRINT"(*Y)(*Y)(*Y)(*Y)(*Y)":CHR$(112)
:CL=1:SE=0
1340 Y=CY+1:GOSUB1080:PRINTCHR$(165);
1350 Z1$=LEFT$(PA$(SU,CA),1):IFZ1$=CHR$(98
)ORZ1$=CHR$(100)THENCL=3:SE=2
1360 COLOR1,CL,SE:PRINTRIGHT$(PA$(SU,CA),2
);
1370 PRINTLEFT$(PA$(SU,CA),1)::COLOR1,1:PR
INT" ";CHR$(167)
1380 Y=CY+2:GOSUB1080:PRINTCHR$(165)::N3$=
" "

```

```

1390 PRINTN3$; :PRINTCHR$(167)
1400 Y=CY+3:GOSUB1080:PRINTCHR$(165); " ";
1410 COLOR1,CL,SE:PRINTLEFT$(PA$(SU,CA),1)
;:COLOR1,1:PRINT" ";CHR$(167)
1420 Y=CY+4:GOSUB1080:PRINTCHR$(165);N3$;C
HR$(167)
1430 Y=CY+5:GOSUB1080:SYS828:PRINTCHR$(165
); " ";
1440 COLOR1,CL,SE:PRINTRIGHT$(PA$(SU,CA),2
); " ";:COLOR1,1:PRINTCHR$(167)
1450 Y=CY+6:GOSUB1080:PRINTCHR$(108);
1460 PRINT"(*P){*P}{*P}{*P}{*P}";:PRINTCHR
$(186)
1470 SOUND1,200,2:RETURN
1480 C=INT(RND(1)*51)+1
1490 IFAV(C)=0THEN1480
1500 AV(C)=0:NC=NC+1:BC(NC)=C:RETURN
1510 REM ** DRAW CARD BACK **
1520 X=CX-1:Y=CY:GOSUB1080:PRINTCHR$(111);
1530 PRINT"(*Y){*Y}{*Y}{*Y}{*Y}";:PRINTCHR
$(112)
1540 CB$=CHR$(165)+"{*+}{*+}{*+}{*+}{*+}" +
CHR$(167)
1550 FORN=1TO5:Y=CY+N:GOSUB1080:PRINTCB$:N
EXT
1560 Y=CY+6:GOSUB1080:PRINTCHR$(108);
1570 PRINT"(*P){*P}{*P}{*P}{*P}";CHR$(186)
1580 SOUND1,300,2:RETURN
1590 REM ** # OF PLAYERS **
1600 SCNCLR:COLOR4,2
1610 X=16:Y=8:GOSUB1080:PRINT"(RED)PONTOON
"
1620 X=12:Y=14:GOSUB1080:INPUT"(LT BLU)1 O
R 2 PLAYERS";NP
1630 IFNP<>1ANDNP<>2THEN1620
1640 FORQ=1TONP:PL(Q)=1:NEXT:RETURN
1650 REM ** DEAL **
1660 FORQ=1TONP
1670 C1=INT(RND(1)*51)+1
1680 IFAV(C1)=0THEN1670
1690 AV(C1)=0:PC(Q,1)=C1:NEXT
1700 PC(Q,1)=C1
1710 C1=INT(RND(1)*51)+1
1720 IFAV(C1)=0THEN1710
1730 AV(C1)=0:BC(1)=C1

```

```
1740 NO(1)=1:NO(2)=1:NC=1:RETURN
1750 REM ** DECODE CARD **
1760 SI=PC(Q,W):CT=1:CD=1
1770 SX=SI:SI=SI-13:IFSI>0THENCD=CD+1:GOTO
1770
1780 SU=CD:CA=SX:BB(Q)=CD:DD(Q)=SX:RETURN
1790 GOTO1770
1800 REM ** SCREEN **
1810 X=0:Y=2:GOSUB1080:PRINT"(BLU)BANK", "P
LAYER 1",
1820 IFNP=2THENPRINT"PLAYER 2"
1830 X=13:Y=20:GOSUB1080:PRINT"(BLK)BET: "
1840 IFNP=2THENX=26:Y=20:GOSUB1080:PRINT"B
ET: "
1850 X=0:Y=21:GOSUB1080:PRINT"(GRN)\":X=13
:Y=21:GOSUB1080:PRINT"\"
1860 X=26:Y=21:GOSUB1080:PRINT"\"
1870 POKE208,0:POKE209,20:SYS828:PRINT"(OR
)RESERVES"
1880 POKE208,0:POKE209,19:SYS828:PRINT"(LT
BLU)TOTAL"
1890 POKE208,13:POKE209,19:SYS828:PRINT"TO
TAL"
1900 IFNP=2THENPOKE208,26:POKE209,19:SYS82
8:PRINT"TOTAL"
1910 RETURN
1920 REM >>>>> DRAW HAND <<<<<<<<
1930 CX=CX(Q):CY=CY(Q)
1940 FORW=1TONO(Q)
1950 GOSUB1760
1960 GOSUB1310
1970 CY=CY+2:CX=CX+1:NEXTW
1980 POKE208,CC(Q)-5:POKE209,21:SYS828:PRI
NTMO(Q)
1990 RETURN
2000 REM * * * STICK * * *
2010 FORQ=1TONP:IFPL(Q)=0THEN2130
2020 CX(Q)=CX(Q)+2:GOSUB2550
2030 X=0:Y=23:GOSUB1080:PRINT"(RED)PLAYER
"Q"(CUR RT)(RVS ON)S(RVS OFF)TICK (RVS ON)
T(RVS OFF)WIST (RVS ON)Q(RVS OFF)UIT (RVS
ON)B(RVS OFF)UY";
2040 GETA$:IFA$="S"ORA$="T"ORA$="Q"ORA$="B
"THEN2050:ELSE2040
2050 CX=CX(Q):CY=CY(Q)+(2*NO(Q))
```

```

2060 IFA$="B"THEN2580
2070 IFA$="Q"THENEND
2080 IFA$="S"ANDSC(Q)>15THENGOSUB2140:CX=C
X+1:GOTO2130
2090 IFA$="T"THENGOSUB2150:CX=CX+1:CX(Q)=C
X(Q)+1
2100 GOSUB2450
2110 IFSC(Q)>21THENX=CC(Q):Y=19:GOSUB1080:
PRINT"(RED)BUST":GOTO2130
2120 IFNO(Q)<>5THEN2040
2130 NEXTQ:RETURN
2140 GOSUB2450:X=CC(Q):Y=19:GOSUB1080:PRIN
T":PRINT"(CUR L)(CUR L)(CUR L)"SC(Q):
:RETURN
2150 C=INT(RND(1)*51)+1
2160 IFAV(C)=0THEN2150
2170 AV(C)=0:NO(Q)=NO(Q)+1
2180 PC(Q,NO(Q))=C:W=NO(Q):GOSUB1760:GOSUB
1310:RETURN
2190 REM ** PLACE BETS **
2200 POKE208,1:POKE209,21:SYS828:PRINTMC;;
FORQ=1TONP
2210 IFPL(Q)=0THEN2310
2220 POKE208,0:POKE209,23:SYS828:FORN=1TO3
8:PRINT":":NEXT
2230 POKE208,0:POKE209,23:SYS828:PRINT"(PU
R)PLAYER ";Q;" YOUR BET (GRN)\\";;
2240 INPUTBE(Q):IFBE(Q)<1THEN2220
2250 IFBE(Q)>MO(Q)THEN2220
2260 POKE208,CC(Q):POKE209,20:SYS828:PRINT
BE(Q)
2270 POKE208,CC(Q):POKE209,21:SYS828:PRINT
"
2280 MO(Q)=MO(Q)-BE(Q)
2290 POKE208,CC(Q)-4:POKE209,21:SYS828:PRI
NT"
2300 POKE208,CC(Q)-5:POKE209,21:SYS828:PRI
NTMO(Q)
2310 NEXTQ
2320 RETURN
2330 REM >>>> DEAL ANOTHER CARD <<<<
2340 C=INT(RND(1)*51)+1
2350 IFAV(C)=0THEN2340
2360 PC(Q,NO(Q)+1)=C:AV(C)=0
2370 NO(Q)=NO(Q)+1

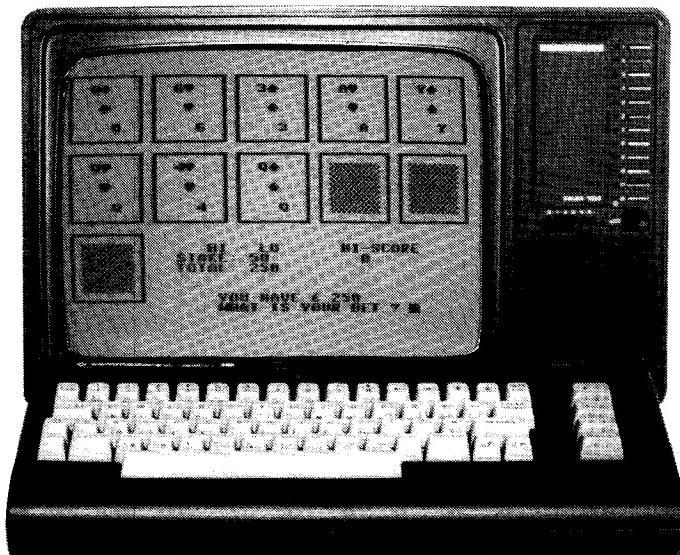
```

```
2380 W=NO(Q):GOSUB1760:GOSUB1310
2390 RETURN
2400 RESTORE2410:FORT=0TO23:READA:POKE1486
4+T,A:NEXT:RETURN
2410 DATA16,56,124,254,124,56,16,0
2420 DATA56,56,254,254,254,254,16,56,0
2430 DATA108,254,254,254,124,56,16,0
2440 REM ** PLAYERS SCORE **
2450 SC(Q)=0:AC=0:FORZ=1TONO(Q):W=Z:COLOR1
,1
2460 GOSUB1760:FC$=RIGHT$(PA$(SU,CA),2)
2470 IFFC$=" Q"ORFC$=" J"ORFC$=" K"ORFC$=
10"THENSC(Q)=SC(Q)+10
2480 IFFC$=" A"THENAC=AC+1:GOTO2500
2490 SC(Q)=SC(Q)+VAL(RIGHT$(PA$(SU,CA),1))
2500 NEXTZ:IFAC>0THENGOSUB2520
2510 X=CC(Q):Y=19:GOSUB1080:PRINT"      ":GOS
UB1080:PRINTSC(Q):RETURN
2520 FORR=1TOAC
2530 IFSC(Q)+11>21THENSC(Q)=SC(Q)+1:ELSESC
(Q)=SC(Q)+11
2540 NEXT:GOTO2510
2550 X=0:Y=23:GOSUB1080:FORA=1TO38:PRINT"
":NEXT:RETURN
2560 FORT=1TO3000:NEXT:RETURN
2570 REM ** BUY **
2580 IFMO(Q)<BE(Q)THEN2100
2590 BE(Q)=BE(Q)*2:MO(Q)=MO(Q)-BE(Q)/2:GOS
UB2150:X=CC(Q):Y=20:GOSUB1080
2600 PRINT"(GRN)      ":GOSUB1080:PRINTBE(Q):
X=X-5:Y=21:GOSUB1080:PRINT"      ":GOSUB1080
2610 PRINTMO(Q):CX=CX+1:CX(Q)=CX(Q)+1:GOTO
2070
2620 REM ** END GAME **
2630 GOSUB2550:X=0:Y=23:GOSUB1080:PRINT"(P
UR)ALL PLAYER'S MONEY GONE"
2640 FORS=100TO500STEP5
2650 SOUND1,S,2:NEXT:GOSUB2560:GOTO60
2660 REM ** WIN SOUND **
2670 FORO=300TO100STEP-10
2680 SOUND1,O,10:NEXT:RETURN
2690 REM ** LOSE SOUND **
2700 SOUND1,ZERO,ZERO
2710 SOUND1,O,10
2720 SOUND2,FIVE,ZERO:RETURN
```

```
2730 RESTORE2740:FORN=0TO8:READA:POKE828+N  
,A:NEXT:RETURN  
2740 DATA24,166,209,164,208,32,240,255,96  
2750 RESTORE2760:FORN=0TO30:READA:POKE882+N,  
A:NEXT:RETURN  
2760 DATA160,0,132,212,132,214,169,208,133  
,213,169,56,133,215,162,3  
2770 DATA177,212,145,214,200,208,249,230,2  
13,230,215,202,16,242,96
```

14

Hi-Lo



Nothing for a Pair

Here is the computer version of the card game featured in a well known TV programme.

Each player starts with £150 and eleven cards are dealt out face down.

The first is turned over and a bet is requested. The bet can be anything up to the player's total holding. When the bet has been placed, 'higher' or 'lower' will be requested. If this is correct then the bet is paid evens. If the guess is wrong or the card is the same value then the stake for that card is lost. This then continues until all the cards have been turned and a new game can then be played.

```
10 REM <<< HI-LO * ANDY/JIM/GREG >>>
20 GOSUB1910:POKE52,55:POKE54,55:POKE56,55
:GOSUB1930:SYS882:VOL8
30 DIMPA(52),AV(52),PA$(5,13),BC(11)
40 DIMCX(11),CY(11)
50 HI=0:GA$="HI-LO":POKE65298,192:POKE6529
9,(PEEK(65299)AND3)OR56
60 COLOR0,2
70 GOSUB1370
80 SCNCLR:COLOR0,2:COLOR4,2:A$=""
90 X=13:Y=17:GOSUB370:PRINT"(RED)HI - LO"
100 GOSUB390
110 GOSUB500
120 GOSUB870
130 X=10:Y=18:GOSUB370:PRINT"(BLU)STAKE:"
140 Y=19:GOSUB370:PRINT"(OR)TOTAL:"
150 X=26:Y=17:GOSUB370:PRINT"(LT BLU)HI-SC
ORE"
160 TT=150
170 BE=0
180 GOSUB1030
190 GOSUB1140
200 C=1
210 CX=CX(C):CY=CY(C)
220 GOSUB1030
230 IFC=11THEN1450
240 PC(1,1)=BC(C):Q=1:W=1
250 GOSUB970:GOSUB580
260 GOSUB1080
270 GOSUB1210
280 GOSUB1030
290 CX=CX(C+1):CY=CY(C+1)
300 PC(1,1)=BC(C+1):CR=CA
310 Q=1:W=1:GOSUB970
320 GOSUB580
330 IFCA>CRANDA$="H"THEN1330:GOTO210
340 IFCA<CRANDA$="L"THEN1330:GOTO210
350 GOTO1280
360 GOTO210
370 POKE208,X:POKE209,Y:SYS828:RETURN
380 REM ** GENERATE PACK **
```

```

390 FORSU=1TO4
400 FORCA=1TO13
410 PA$(SU,CA)=CHR$(SU+96)
420 IFCA<11ANDCA>1THENPA$(SU,CA)=PA$(SU,CA)
) +CHR$(32)+CHR$(CA+48)
430 IFCA=1THENPA$(SU,CA)=PA$(SU,CA)+" A"
440 IFCA=11THENPA$(SU,CA)=PA$(SU,CA)+" J"
450 IFCA=10THENPA$(SU,CA)=PA$(SU,CA)+" 10"
460 IFCA=12THENPA$(SU,CA)=PA$(SU,CA)+" Q"
470 IFCA=13THENPA$(SU,CA)=PA$(SU,CA)+" K"
480 NEXT:NEXT:RETURN
490 REM ** SHUFFLE **
500 FORN=1TO52:PA(N)=N:AV(N)=1:NEXT
510 FORA=1TO30
520 N1=INT(RND(1)*51)+1:N2=INT(RND(1)*51)+1
530 CC=PA(N1)
540 PA(N1)=PA(N2)
550 PA(N2)=CC
560 NEXT:RETURN
570 REM ** DRAW CARD **
580 X=CX-1:Y=CY:GOSUB370
590 COLOR1,1:PRINTCHR$(111);
600 PRINT"(*Y)(*Y)(*Y)(*Y)(*Y)":CHR$(112):
CL=1:SE=0
610 Y=CY+1:GOSUB370:PRINTCHR$(165);
620 Z1$=LEFT$(PA$(SU,CA),1):IFZ1$=CHR$(98)
ORZ1$=CHR$(100)THENCL=3:SE=2
630 COLOR1,CL,SE:PRINTRIGHT$(PA$(SU,CA),2)
;
640 PRINTLEFT$(PA$(SU,CA),1)::COLOR1,1:PRIORITY" ";CHR$(167)
650 Y=CY+2:GOSUB370:PRINTCHR$(165)::N3$=
"
660 PRINTN3$::PRINTCHR$(167)
670 Y=CY+3:GOSUB370:PRINTCHR$(165)::" ";
680 COLOR1,CL,SE:PRINTLEFT$(PA$(SU,CA),1)::COLOR1,1:PRINT" ";CHR$(167)
690 Y=CY+4:GOSUB370:PRINTCHR$(165)::N3$;CHR$(167)
700 Y=CY+5:GOSUB370:SYS828:PRINTCHR$(165)::" ";
710 COLOR1,CL,SE:PRINTRIGHT$(PA$(SU,CA),2)::" ";
COLOR1,1:PRINTCHR$(167)
720 Y=CY+6:GOSUB370:PRINTCHR$(108);

```

```

730 PRINT"(*P)(*P)(*P)(*P)(*P)":PRINTCHR$  

(186)  

740 SOUND1,200,2:RETURN  

750 C=INT(RND(1)*51)+1  

760 IFAV(C)=0THEN750  

770 AV(C)=0:NC=NC+1:BC(NC)=C:RETURN  

780 REM ** DRAW CARD BACK **  

790 X=CX-1:Y=CY:GOSUB370:PRINTCHR$(111);  

800 PRINT"(*Y}{*Y}{*Y}{*Y}{*Y)":PRINTCHR$  

(112)  

810 CB$=CHR$(165)+"{++}{++}{++}{++}{++}"+C  

HR$(167)  

820 FORN=1TO5:Y=CY+N:GOSUB370:PRINTCB$:NEXT  

830 Y=CY+6:GOSUB370:PRINTCHR$(108);  

840 PRINT"(*P)(*P)(*P)(*P)(*P)":CHR$(186)  

850 SOUND1,300,2:RETURN  

860 REM >>>>> DRAW CARDS <<<<<<  

870 RESTORE930:FORA=1TO11  

880 READCX(A),CY(A)  

890 CX=CX(A):CY=CY(A)  

900 GOSUB790  

910 NEXTA  

920 RETURN  

930 DATA1,0,9,0,17,0,25,0,33,0  

940 DATA1,8,9,8,17,8,25,8,33,8  

950 DATA1,16  

960 REM >>>> DECODE CARD <<<<<<  

970 SI=PC(Q,W):CT=1:CD=1  

980 SX=SI:SI=SI-13:IFSI>0THENCD=CD+1:GOTO9  

80  

990 SU=CD:CA=SX:RETURN  

1000 GOTO980  

1010 RETURN  

1020 REM >> PRINT HI TOTAL BET <<  

1030 COLOR1,6,2:X=16:Y=18:GOSUB370:PRINT"  

":GOSUB370:PRINTBE;  

1040 Y=19:GOSUB370:PRINT"      ":GOSUB370:P  

RINTTT  

1050 X=27:Y=18:GOSUB370:PRINT"      ":GOSUB  

370:PRINTHI  

1060 RETURN  

1070 REM >>>>>> PLACE BETS <<<<<<  

1080 X=14:Y=22:GOSUB370:PRINT"(RED)YOU HAV  

E \";TT

```

```

1090 Y=23:GOSUB370:INPUT"(BLK)WHAT IS YOUR
   BET ";BE
1100 IFBE<10RBE>TTTHEN1080
1110 TT=TT-BE:GOSUB1030
1120 RETURN
1130 REM >>>>> GENERATE CARDS <<<<<
1140 FORA=1TO11
1150 C=INT(RND(1)*51)+1
1160 IFAV(C)=ØTHEN1150
1170 BC(A)=PA(C)
1180 AV(C)=Ø
1190 NEXTA:RETURN
1200 REM >>>>>> HIGHER LOWER <<<<<<
1210 X=14:Y=23:GOSUB370:FORB=1TO25:PRINT"
";:NEXTB
1220 X=14:Y=22:GOSUB370:FORA=1TO20:PRINT"
";:NEXTA
1230 X=14:Y=23:GOSUB370:PRINT"(BRN)HIGHER
/ LOWER"
1240 GETA$
1250 IFA$<>"H"ANDA$<>"L"THEN1240
1260 RETURN
1270 REM >>>>>>>>> LOSE <<<<<<<<<
1280 SOUND1,300,20:SOUND1,0,10
1290 SOUND1,500,40
1300 IFTT<1THEN1490
1310 C=C+1:GOT0210
1320 REM >>>>>>>>> WIN <<<<<<<<<<
1330 FORS=500TO300STEP-10
1340 SOUND1,S,2:NEXTS
1350 TT=TT+BE*2
1360 C=C+1:GOT0210
1370 RESTORE1380:FORT=ØTO23:READA:POKE1486
4+T,A:NEXT:RETURN
1380 DATA16,56,124,254,124,56,16,Ø
1390 DATA56,56,254,254,254,16,56,Ø
1400 DATA108,254,254,254,124,56,16,Ø
1410 REM
1420 RETURN
1430 FORT=1TO3000:NEXT:RETURN
1440 REM >>>>>> HIGH SCORES <<<<<<<
1450 IFTT>HITHENHI=TT
1460 SC=TT:GOSUB1580
1470 GOT080
1480 REM >>>>>> NO MONEY LEFT <<<<<<

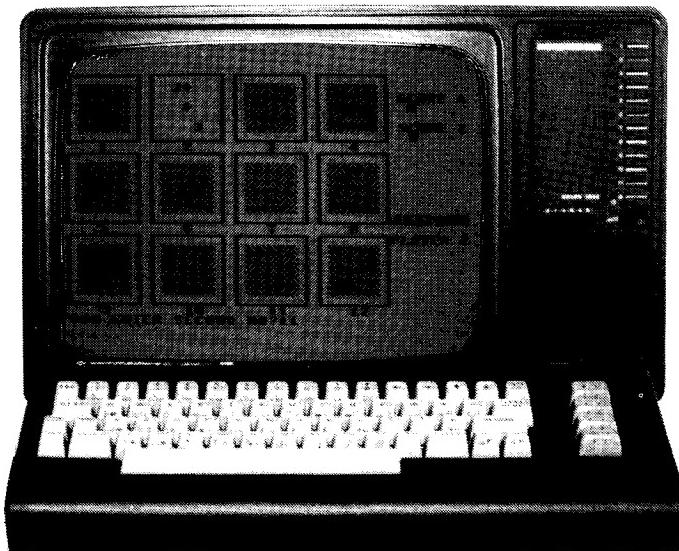
```

```
1490 X=14:Y=23:GOSUB370:FORA=1TO25:PRINT"
";:NEXTA
1500 X=14:Y=23:GOSUB370:PRINT"(PUR)RUN OUT
OF MONEY !"
1510 GOSUB1430:GOT080
1520 LE=0:RI=0:FI=0
1530 KS=PEEK(198)
1540 IFKS=10THENLE=1
1550 IFKS=13THENRI=1
1560 IFKS=60THENFI=1
1570 RETURN
1580 RR=0:GOSUB1870
1590 FORA=5TO1STEP-1
1600 IFSC>SC(A)ANDSC<SC(A-1)THENRR=A
1610 NEXT
1620 IFRR=0THEN1850
1630 FORA=4TORRSTEP-1:NA$(A+1)=NA$(A):SC(A
+1)=SC(A):NEXT
1640 NA$(RR)=""":SC(RR)=SC
1650 GOSUB1870
1660 X=6:Y=12:GOSUB370:PRINT"(LT BLU)PLEAS
E ENTER YOUR NAME "
1670 N$="ABCDEFGHIJKLMNOPQRSTUVWXYZ. ]"
1680 X=4:Y=14:GOSUB370:PRINTN$:XX=4
1690 X=0:Y=16:GOSUB370:PRINT" ] =
END."
1700 X=XX:Y=15:GOSUB370:PRINT"-"
1710 X=9:Y=21:PRINTNA$(RR)
1720 GOSUB1520
1730 IFLE=1ANDXX>4THENX=XX:Y=15:GOSUB370:P
RINT" " :XX=XX-1
1740 IFRI=1ANDXX<32THENX=XX:Y=15:GOSUB370:
PRINT" " :XX=XX+1
1750 IFFI=1ANDMID$(N$, (XX-3), 1)=" ] " THEN179
0
1760 IFFI=1THENNA$(RR)=NA$(RR)+MID$(N$, (XX
-3), 1)
1770 IFLEN(NA$(RR))=15THEN1790
1780 GOT01700
1790 GOSUB1870
1800 X=7:Y=16:GOSUB370:PRINT"(BLK)PRESS SP
ACE TO PLAY AGAIN."
1810 KS=PEEK(198)
1820 IFKS<>60THEN1810
1830 GETA$
```

```
1840 FORT=0TO8:POKE1319+T,0:NEXT:RETURN
1850 IFSC<SC(1)THEN1790
1860 RR=1:GOT01630
1870 SCNCLR:X=6:Y=1:GOSUB370:PRINT"(PUR)HI
GH SCORES FOR "GA$
1880 COLOR1,3,2:X=3:FORA=1TO5
1890 Y=A+4:GOSUB370:PRINTA;"....";NA$(A);T
AB(30)SC(A)
1900 NEXT:RETURN
1910 RESTORE1920:FORN=0TO8:READA:POKE828+N
,A:NEXT:RETURN
1920 DATA24,166,209,164,208,32,240,255,96
1930 RESTORE1940:FORN=0TO30:READA:POKE882+
N,A:NEXT:RETURN
1940 DATA160,0,132,212,132,214,169,208,133
,213,169,56,133,215,162,3
1950 DATA177,212,145,214,200,208,249,230,2
13,230,215,202,16,242,96
```

15

Concentration



Find the Pair

This deceptively simple game requires a good memory and of course great concentration.

Each player has one turn to select two cards. If they match then they score and are removed from the screen. The next player then tries to find a pair until all the cards are gone. The winner is the one who correctly matched the most pairs.

In some card versions of this game a successful player is allowed to turn over another two cards. However, the game plays better in this 12-card version if one player cannot clear the decks so quickly.

Here is a money raising tip. Those readers who are involved in raising cash for charities or school funds may like to consider how the micro can help. *Concentration* could be used as a fund raiser, just as many other programs featured in the book can. Simply set up your computer and charge visitors for the privilege of pressing the buttons!

```
10 REM << CONCENTRATION ANDY/JIM/GREG >>
20 GOSUB1830:POKE52,55:POKE54,55:POKE56,55
:GOSUB1850:SYS882:VOL8
30 POKE65298,192:POKE65299,(PEEK(65299)AND
3)OR56
40 CLR:XX=RND(-TI)
50 DIMPA$(5,13),CC(3),SC(2)
60 DIMPA(52),AV(52)
70 DIMCX(12),CY(12)
80 DIMCA(12),BC(12)
90 DIMPC(1,2),AC(12)
100 GOSUB1690
110 SCNCLR:GOSUB270
120 COLOR0,2:COLOR4,2
130 SC(1)=0:SC(2)=0
140 GOSUB270
150 GOSUB380
160 GOSUB800
170 GOSUB990
180 FORA=1TO12:AC(A)=1:NEXTA
190 P=1
200 GOSUB1440
210 GOSUB1170
220 GOT01330
230 IFP=1THENP=2:ELSEP=1
240 GOT0200
250 POKE208,X:POKE209,Y:SYS828:RETURN
260 REM ** GENERATE PACK **
270 FORSU=1TO4
280 FORCA=1TO13
290 PA$(SU,CA)=CHR$(SU+96)
300 IFCA<11ANDCA>1THENPA$(SU,CA)=PA$(SU,CA)
)+CHR$(32)+CHR$(CA+48)
310 IFCA=1THENPA$(SU,CA)=PA$(SU,CA)+" A"
320 IFCA=11THENPA$(SU,CA)=PA$(SU,CA)+" J"
330 IFCA=10THENPA$(SU,CA)=PA$(SU,CA)+" 10"
340 IFCA=12THENPA$(SU,CA)=PA$(SU,CA)+" Q"
350 IFCA=13THENPA$(SU,CA)=PA$(SU,CA)+" K"
360 NEXT:NEXT:RETURN
370 REM ** SHUFFLE **
380 FORN=1TO52:PA(N)=N:AV(N)=1:NEXT
390 FORA=1TO30
400 N1=INT(RND(1)*51)+1:N2=INT(RND(1)*51)+
1
410 CC=PA(N1)
```

```

420 PA(N1)=PA(N2)
430 PA(N2)=CC
440 NEXT:RETURN
450 REM ** DRAW CARD **
460 X=CX-1:Y=CY:GOSUB250
470 COLOR1,1:PRINTCHR$(111);
480 PRINT"(*Y)(*Y)(*Y)(*Y)(*Y)";CHR$(112):
CL=1:SE=0
490 Y=CY+1:GOSUB250:PRINTCHR$(165);
500 Z1$=LEFT$(PA$(SU,CA),1):IFZ1$=CHR$(98)
ORZ1$=CHR$(100) THEN CL=3:SE=2
510 COLOR1,CL,SE:PRINTRIGHT$(PA$(SU,CA),2)
;
520 PRINTLEFT$(PA$(SU,CA),1)::COLOR1,1:PRT
NT" ";CHR$(167)
530 Y=CY+2:GOSUB250:PRINTCHR$(165)::N3$=
"
540 PRINTN3$::PRINTCHR$(167)
550 Y=CY+3:GOSUB250:PRINTCHR$(165);" ";
560 COLOR1,CL,SE:PRINTLEFT$(PA$(SU,CA),1);
:COLOR1,1:PRINT" ";CHR$(167)
570 Y=CY+4:GOSUB250:PRINTCHR$(165);N3$;CHR
$(167)
580 Y=CY+5:GOSUB250:SYS828:PRINTCHR$(165);
" ";
590 COLOR1,CL,SE:PRINTRIGHT$(PA$(SU,CA),2)
;" ";COLOR1,1:PRINTCHR$(167)
600 Y=CY+6:GOSUB250:PRINTCHR$(108);
610 PRINT"(*P){*P}{*P}{*P}{*P}";:PRINTCHR$(
186)
620 SOUND1,200,2:RETURN
630 C=INT(RND(1)*51)+1
640 IFAV(C)=0 THEN 630
650 AV(C)=0:NC=NC+1:BC(NC)=C:RETURN
660 REM ** DRAW CARD BACK **
670 COLOR1,7,2:X=CX-1:Y=CY:GOSUB250:PRINTC
HR$(111);
680 PRINT"(*Y)(*Y)(*Y)(*Y)(*Y)";:PRINTCHR$(
112)
690 CB$=CHR$(165)+"(+){+}{+}{+}{+}{+}"+C
HR$(167)
700 FORN=1TO5:Y=CY+N:GOSUB250:PRINTCB$:NEX
T
710 Y=CY+6:GOSUB250:PRINTCHR$(108);
720 PRINT"(*P){*P}{*P}{*P}{*P}";CHR$(186)

```

```
730 SOUND1,300,2:RETURN
740 REM >>>>> ERASE CARD <<<<<<
750 X=CX-1:Y=CY:GOSUB250
760 FOR YD=1 TO 7
770 Y=CY+YD-1:GOSUB250:PRINT"
780 NEXT YD:RETURN
790 REM >>>>> GENERATE CARDS <<<<<<
800 RESTORE 940:FOR A=1 TO 12
810 CA(A)=1:NEXT A
820 FOR A=1 TO 12 STEP 2
830 READ C1,C2
840 C=INT(RND(1)*12)+1
850 IF CA(C)=0 THEN 840
860 BC(C)=C1:CA(C)=0
870 C=INT(RND(1)*12)+1
880 IF CA(C)=0 THEN 870
890 BC(C)=C2:CA(C)=0
900 NEXT A
910 FOR A=1 TO 12
920 READ CX(A),CY(A)
930 NEXT A:RETURN
940 DATA 1,14,2,15,3,16,4,17,5,18,6,19
950 DATA 1,0,9,0,17,0,25,0
960 DATA 1,8,9,8,17,8,25,8
970 DATA 1,16,9,16,17,16,25,16
980 REM >>>>> DRAW SCREEN <<<<<<
990 COLOR 1,7,2
1000 FOR A=1 TO 12
1010 CX=CX(A):CY=CY(A)
1020 GOSUB 670
1030 X=CX(A)+1:Y=CY(A)+7:GOSUB250:PRINT A
1040 NEXT A
1050 X=32:Y=2:GOSUB250:PRINT"(LT BLU)SCORE
   1"
1060 Y=5:GOSUB250:PRINT"(BLK)SCORE 2"
1070 X=31:Y=14:GOSUB250:PRINT"(LT BLU)PLAY
ER 1";
1080 Y=16:GOSUB250:PRINT"(BLK)PLAYER 2"
1090 RETURN
1100 REM >>>>> DECODE CARD <<<<<<
1110 SI=PC(Q,W):CT=1:CD=1
1120 SX=SI:SI=SI-13:IF SI>0 THEN CD=CD+1:GOTO
1120
1130 SU=CD:CA=SX:RETURN
1140 GOTO 1120
```

```

1150 RETURN
1160 REM >>>>> INPUT NUMBER <<<<<<
1170 X=0:Y=24:GOSUB250:FORA=1TO30:PRINT" "
;:NEXT
1180 GOSUB250:PRINT"(RED)ENTER FIRST NO?";
:GOSUB1760:N1=U
1190 IFN1<10RN1>12THEN1170
1200 IFAC(N1)=0THEN1180
1210 PC(1,1)=BC(N1):Q=1:W=1
1220 GOSUB1110:CX=CX(N1):CY=CY(N1)
1230 GOSUB460
1240 X=0:Y=24:GOSUB250:FORA=1TO30:PRINT" "
;:NEXT
1250 X=0:Y=24:GOSUB250:PRINT"(RED)NOW ENTE
R SECOND NO?";:GOSUB1760:N2=U
1260 IFN2<10RN2>12THEN1240
1270 IFAC(N2)=0THEN1240
1280 IFN1=N2THEN1240
1290 PC(1,2)=BC(N2):Q=1:W=2
1300 GOSUB1110:CX=CX(N2):CY=CY(N2)
1310 GOSUB460:RETURN
1320 REM >>>>> CHECK PAIR <<<<<<
1330 Q=1:W=1:GOSUB1110
1340 CO=CA:Q=1:W=2:GOSUB1110
1350 CP=CA
1360 IFCO=CPTHEN1500
1370 X=0:Y=24:GOSUB250:FORA=1TO30:PRINT" "
;:NEXT
1380 GOSUB250:PRINT"(OR)CARDS DO NOT MATCH
";
1390 GOSUB1750
1400 CX=CX(N1):CY=CY(N1)
1410 GOSUB670:CX=CX(N2):CY=CY(N2)
1420 GOSUB670:GOTO230
1430 REM >>>>> PRINT SCORES <<<<<<
1440 COLOR1,6,2:X=32:Y=3:GOSUB250:PRINTSC(
1)
1450 Y=6:GOSUB250:PRINTSC(2)
1460 IFP=1THENP1$="{}{RVS ON}{LT BLU}PLAYER
1{}{RVS OFF}":P2$="{}{BLK}PLAYER 2"
1470 IFP=2THENP1$="{}{LT BLU}PLAYER 1":P2$="{}{BLK}{}{RVS ON}PLAYER 2{}{RVS OFF}"
1480 X=31:Y=14:GOSUB250:PRINTP1$;:Y=16:GOS
UB250:PRINTP2$;:RETURN
1490 REM >>>>> PAIR FOUND <<<<<<<
```

```
1500 X=0:Y=24:GOSUB250:FORA=1TO30:PRINT" "
;:NEXT
1510 GOSUB250:PRINT"(RED)PAIR FOUND";
1520 AC(N1)=0:AC(N2)=0
1530 FORS=500TO100STEP-10:SOUND1,S,1:NEXTS
1540 GOSUB1750
1550 CX=CX(N1):CY=CY(N1)
1560 GOSUB750:CX=CX(N2):CY=CY(N2)
1570 GOSUB750:SC(P)=SC(P)+5
1580 IFSC(1)+SC(2)=30THEN1610
1590 GOT0230
1600 REM >>>>> END OF GAME <<<<<<
1610 SCNCLR:COLOR1,3,2
1620 X=3:Y=2:GOSUB250:PRINT"PLAYER 1 SCORE
:"SC(1)
1630 Y=4:GOSUB250:PRINT"PLAYER 2 SCORE:"SC
(2)
1640 X=7:Y=11:GOSUB250:PRINT"(BLK)PRESS SP
ACE TO PLAY AGAIN"
1650 X=7:Y=13:GOSUB250:PRINT"(LT BLU)OR 'Q
' TO QUIT."
1660 GETA$:IFA$==" "THEN40
1670 IFA$=="Q"THENSYS32768
1680 GOT01660
1690 RESTORE1700:FORT=0TO23:READA:POKE1486
4+T,A:NEXT:RETURN
1700 DATA16,56,124,254,124,56,16,0
1710 DATA56,56,254,254,254,16,56,0
1720 DATA108,254,254,254,124,56,16,0
1730 REM
1740 RETURN
1750 FORT=1TO3000:NEXT:RETURN
1760 D$==" "
1770 GETKEYA$
1780 IFASC(A$)=13THEN1800
1790 D$=D$+A$:PRINTA$::GOT01770
1800 U=VAL(D$):IFU<10RU>12THEN1820
1810 RETURN
1820 FORE=1TOLEN(D$):PRINT"(CUR L)";:NEXT:
GOT01760
1830 RESTORE1840:FORN=0TO8:READA:POKE828+N
,A:NEXT:RETURN
1840 DATA24,166,209,164,208,32,240,255,96
1850 RESTORE1860:FORN=0TO30:READA:POKE882+N
,A:NEXT:RETURN
```

1860 DATA160,0,132,212,132,214,169,208,133
,213,169,56,133,215,162,3
1870 DATA177,212,145,214,200,208,249,230,2
13,230,215,202,16,242,96

16

Micro Mind



Colour Your Reasoning

Versions of this well known and widely enjoyed game can be found on all microcomputers. In writing this implementation of the game, care has been taken to produce a good looking layout that makes play enjoyable.

The object of the game is to work out what the secret colour combination is, by placing coloured markers in a row.

The Commodore 16 will then indicate how accurate the placing was:

The right colour, right place is indicated by a HEART.

The wrong colour, wrong place is indicated by a CROSS.

The right colour, wrong place is indicated by an ASTERISK.

S and A are used to select the colours by moving an arrow to the colour required. The space bar then places it. The next ones are then entered in the same manner until the row is full and the result is shown.

There are ten chances to guess the correct sequence before each game ends. The computer will then display the correct combination at the top of the screen.

The game can be played competitively by each person aiming to achieve a lower number of lines to work out the combinations.

```

10 REM >>>>>>MICRO MIND<<<<<
20 REM >>>>GREG/ANDY/JIM<<<<<
30 FORT=1TO6:READP(T),S(T):NEXT:VOL8:COLOR
0,1:COLOR4,1:GRAPHIC1
40 SCNCLR:GOSUB170:G=25:H=24:Z$=" " :FORA=
1TO4:X=(A*4)-2:B$="(^Q)(^W)"
50 CHAR1,X,2,Z$,1:NEXT:FORY=4TO22STEP2:FOR
X=2TO16STEP4:CHAR1,X,Y,B$:NEXTX,Y
60 L=1:GOSUB90
70 GOSUB100:GOSUB190:IFA$="YYYY"THEN300
80 L=L+1:IFL=11THEN290:ELSEGOTO70
90 FORO=1TO4:N=INT(RND(1)*6)+1:N(O)=N:NEXT
:RETURN
100 FORC=1TO6:COLOR1,P(C),S(C):X=C+24:CHAR
1,X,4," ",1:NEXT:Q=2
110 COLOR1,2,7:CHAR1,G,5,"^":FORT=1TO20:NE
XT:CHAR1,G,5," "
120 GOSUB350:IFE=1ANDG>25THENG=G-1
130 IFR=1ANDG<30THENG=G+1:ELSEIFF=1THEN150
140 GOTO110
150 U=G-24:COLOR1,P(U),S(U):CHAR1,Q,L*2+2,
"({^Q})":Q=Q+4:U(Q/4)=U
160 IFQ=18THENRETURN:ELSEGOTO110
170 RESTORE380:COLOR1,2,7:CHAR1,4,0,"MICRO
MIND"
180 DRAW1,200,189:FORT=1TO43:READX,Y:DRAWT
0X,Y:NEXT:RETURN
190 A$="":Q=3:COLOR1,9,5:FORZ=1TO4
200 IFU(Z)=N(Z)THENCHAR1,Q,L*2+2,"({^S})":A$=
=A$+"Y":GOTO220
210 CHAR1,Q,L*2+2,"({^V})":A$=A$+"X"
220 Q=Q+4:NEXT:Q=3:S=0:FORZ=1TO4
230 IFMID$(A$,Z,1)="Y"THEN SOUND1,100,10:GO
TO270
240 FORW=1TO4
250 IFU(Z)=N(W)ANDMID$(A$,W,1)="X"THENCHAR
1,Q,L*2+2,"*":GOSUB280
260 NEXT:IFS=0THEN SOUND1,450,10
270 Q=Q+4:SOUND1,0,5:NEXT:RETURN

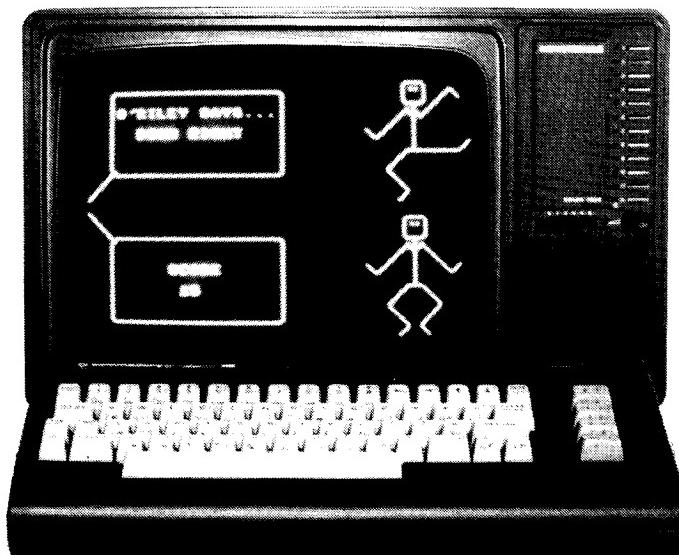
```

```
280 SOUND1,300,10:S=1:RETURN
290 M$="SORRY":GOTO310
300 M$="DID IT IN "+STR$(L)+" TRYS"
310 FOR A=1 TO 4:M=N(A):COLOR1,P(M),S(M):X=(A
#4)-2:CHAR1,X,2,Z$,1:NEXT
320 CHAR1,23,6,M$
330 CHAR1,24,15,"TRY AGAIN"
340 GOSUB350: IFF=0 THEN 340: ELSE GOTO 40
350 E=0:R=0:F=0:K=PEEK(198): IF K=10 THEN E=1
360 IF K=13 THEN R=1:ELSE IF K=60 THEN F=1
370 RETURN: DATA 3,2,5,4,6,3,8,7,4,5,7,3
380 DATA 206,177,202,173,192,172,189,168,18
9,166,193,159,189,156,187,152,190
390 DATA 150,188,148,186,146,188,143,190,14
1,191,138,191,135,189,133,179,133
400 DATA 176,130,176,128,178,126,188,114,19
0,106,192,97,200,87,212,79,230,75
410 DATA 238,74,250,77,262,81,271,90,276,98
,278,108,278,114,276,122,275,132
420 DATA 272,142,270,150,267,160,264,164,26
0,168,254,174,255,177,270,189
```

300
370

17

O'Grady Says



O'Grady Says: Type This In

This is an implementation of the well known game, sometimes also known as Simon Says.

This program can be most helpful in teaching the concept of left and right, but be warned, as it stands the program means *its* left or right, not yours!

You start each game with just one point. The speech bubble will then announce the action required together with the name of the person giving the order. If O'Grady says 'do it' then you do it fast! If O'Riley says 'do it' then you do *not*! The little chap at the top follows all commands.

There are four possible actions which you may be asked to do. Up and Down are actioned using the up or down cursor as normal. Left or Right are obtained using the respective key in relation to the on-screen character. This means that pressing the left cursor will actually lift up the chap's right arm.

This swap has been incorporated to provide more of a mental challenge. It can be quite a struggle to work out which is left, which is

right and of course whether O'Grady said 'do it'. Play the game and you will see what I mean.

Speedy mental processing is important in *O'Grady Says* or lack of action may be taken as intentional. This could mean that you lose points or, if you are lucky, doing nothing could be just the right response.

```
240 COLOR1,2,6
250 IFOG=1THENCHAR1,4,3,"O'GRADY SAYS... "
260 IFOG=2THENCHAR1,4,3,"O'RILEY SAYS... "
270 GOSUB70
280 COLOR1,2,7
290 IFPO=1THENCHAR1,4,5,"MOVE UP"
300 IFPO=2THENCHAR1,4,5,"MOVE DOWN"
310 IFPO=3THENCHAR1,4,5,"UP RIGHT"
320 IFPO=4THENCHAR1,4,5,"UP LEFT"
330 T=0:BM=0
340 GETKE$
350 IFKE$="({CUR UP})"THENBM=1:GOTO400
360 IFKE$="({CUR DN})"THENBM=2:GOTO400
370 IFKE$="({CUR RT})"THENBM=3:GOTO400
380 IFKE$="({CUR L})"THENBM=4:GOTO400
390 T=T+1:IFT=40THEN450:ELSE340
400 GOSUB80:COLOR1,2,7
410 IFPO=1ANDBM=1ANDOG=1THEN620
420 IFPO=2ANDBM=2ANDOG=1THEN620
430 IFPO=3ANDBM=3ANDOG=1THEN620
440 IFPO=4ANDBM=4ANDOG=1THEN620
450 IFT=40ANDOG=2THENSC=SC+1:GOSUB790:GOTO
510
460 IFOG=2ANDKE$(< >)"THENGOSUB810:GOT0490
470 IFT=40ANDOG=1THENGOSUB830:GOT0490
480 GOSUB760:CHAR1,8,5,"WRONG !":SOUND1,20
8,10:SOUND1,0,20
490 SC=SC-1
500 IFSC<1THEN550
510 GOSUB650
520 FORT=0TO999:NEXT
530 GOTO200
540 REM >>>>END OF GAME <<<<
550 CHAR1,4,17,"BAD LUCK, YOU"
560 CHAR1,4,18,""
570 CHAR1,4,19,"MADE IT THROUGH"
580 CHAR1,4,20,""
590 Q$=STR$(TU)+" TURNS."
600 CHAR1,4,21,Q$
610 FORT=1TO2000:NEXT:RUN
620 GOSUB760:CHAR1,7,5,"CORRECT !"
630 FORR=600TO800STEP10:SOUND1,R,2:NEXT
640 SC=SC+1:GOSUB650:GOT0510
650 Q$=" ({CUR L}) ({CUR L}) ({CUR L}) ({CUR L}) "+STR$(SC):CHAR1,10,20,Q$::RETURN
```



```
{CUR L}{^--}
730 A$(4)=A$(4)+" {CUR DN}{CUR L}{CUR L}{^N
}{{*Y}}{{*Y}}{{*Y}}{{*Y}}{CUR UP}{^N}{CUR DN}{^
CUR DN}{CUR L}{CUR L}{CUR L}{CUR L}{CUR L}{^
CUR L}{CUR L}{CUR L}{^N}{CUR DN}{CUR L}{^N}{HOME}"
740 CL$="
750 RETURN
760 FORR=3T07STEP2:CHAR1,4,R,""
":NEXT:RETURN
770 FORR=0T011:CHAR1,26,R,CL$:NEXT:RETURN
780 FORR=13T024:CHAR1,26,R,CL$:NEXT:RETURN
790 GOSUB760:CHAR1,4,5,"WELL DONE !":CHAR1
,4,7,"O'RILEY SAID IT"
800 SOUND1,200,10:SOUND1,400,10:RETURN
810 GOSUB760:CHAR1,4,5,"WRONG !":CHAR1,4,7
,"O'RILEY SAID IT"
820 SOUND3,900,10:SOUND3,999,10:RETURN
830 GOSUB760:CHAR1,5,5,"OUT OF TIME !"
840 SOUND1,200,10:SOUND1,0,20:RETURN
```

18

Parrot



Who's a Pretty Program?

Here are four pretty parrots, each a different colour and each holding a 'string' in his mouth.

When the program runs the parrots will commence pulling the strings in sequence. The first set will consist of only one pull. Then it will be your turn to copy the sequence.

- The perches of the birds have been arranged to correspond to the arrangement of the keys: Cursor up = the top parrot; * = the bottom parrot; A = the left parrot; S = the right parrot.

The computer produces a sequence which starts off simply but gets very long. During testing of this program no one was able to remember more than a sequence of 25. Amnesia usually sets in at around 12. As it stands the longest sequence that the parrots will produce is 50 long. If you remember that length then you must have some sort of photographic memory – or perhaps you are an android!

This simple game can be great fun at parties where all sorts of forfeits or penalties may be devised for the losers.

```
10 REM <<<< PARROT >>>> ANDY/JIM/GREG >>>
20 POKE52,55:POKE54,55:POKE56,55:GOSUB1300
:GOSUB1340:SYS882:VOL8
30 POKE65298,192:POKE65299,(PEEK(65299)AND
3)OR56
40 DIMNO(50):XX=RND(-TI)
50 COLOR0,1:COLOR4,1:COLOR1,8,5:SCNCLR
60 GOSUB520:GOSUB840
70 SC=0:HI=0
80 GOSUB1030
90 RESTORE100
100 DATA3,8,4,6
110 FORA=1TO4:C=A:READC:COLOR1,C,4:GOSUB96
0
120 NEXT
130 P0=0
140 REM >>>>>>>> MAIN LOOP <<<<<<<<<
150 N=INT(RND(1)*4)+1
160 P0=P0+1
170 NO(P0)=N
180 GOSUB1200:X=1:Y=21:GOSUB510:PRINT"(BLU
){YEL}MY TURN"
190 FORB=1TOPO
200 GOSUB1140
210 NEXT
220 FORT=1TO1000:NEXT
230 B=1
240 GOSUB1200:X=1:Y=21:GOSUB510:PRINT"(LT
BLU)YOUR TURN"
250 GOSUB1230
260 IFLE=1THEN A=4:SOUND1,800,10:COLOR1,6,5
:GOSUB1000:FORT=1TO500:NEXT:GOT0310
270 IFRI=1THEN A=2:SOUND1,400,10:COLOR1,8,5
:GOSUB1000:FORT=1TO500:NEXT:GOT0310
280 IFUP=1THEN A=1:SOUND1,200,10:COLOR1,3,5
:GOSUB1000:FORT=1TO500:NEXT:GOT0310
290 IFDN=1THEN A=3:SOUND1,600,10:COLOR1,4,5
:GOSUB1000:FORT=1TO500:NEXT:GOT0310
300 GOT0250
310 GOSUB960
320 IFNO(B)=ATHEN330:ELSE360
330 IFB=POTHENGOSUB1200:X=1:Y=21:GOSUB510:
PRINT"(RED)WELL DONE,"P0" SO FAR."
340 IFB=POTHENFORT=1TO1500:NEXT:GOSUB1200:
GOT0150
```

```
350 B=B+1:GOTO250
360 GOSUB1200
370 X=1:Y=21:GOSUB510:PRINT"(PUR)WRONG SEQ
UENCE"
380 Y=22:GOSUB510:PRINT"AFTER "PO-1" MOVEM
ENTS."
390 FORT=1TO2000:NEXT
400 GOSUB1200
410 Y=21:GOSUB510:PRINT"(GRN)SEQUENCE WAS"
420 FORT=1TO1000:NEXT
430 FORB=1TOPO
440 GOSUB1140
450 NEXT
460 GOSUB1200
470 X=1:Y=21:GOSUB510:PRINT"(CYN)TRY AGAIN
.
480 FORT=1TO2000:NEXT
490 GOSUB1200
500 GOTO130
510 POKE208,X:POKE209,Y:SYS828:RETURN
520 RESTORE530:FORT=0TO239:READA:POKE14848
+T,A:NEXT:RETURN
530 DATA2,2,2,2,0,7,15,31
540 DATA0,0,3,7,7,206,254,255
550 DATA0,0,224,248,248,124,124,252
560 DATA31,25,16,0,1,2,4,0
570 DATA255,63,143,135,7,7,15,31
580 DATA252,248,240,240,240,248,252,254
590 DATA30,29,61,59,59,59,61,30
600 DATA255,255,255,255,255,255,255,63
610 DATA0,128,192,224,240,248,252,254
620 DATA15,7,0,0,0,28,63,0
630 DATA199,248,255,63,48,96,248,0
640 DATA255,63,255,255,31,3,0,0
650 DATA0,128,192,224,240,248,124,3
660 DATA2,2,2,2,0,30,127,127
670 DATA0,0,0,0,0,0,0,192
680 DATA0,0,0,0,0,0,0,7
690 DATA0,0,0,0,0,15,127,255
700 DATA0,0,1,31,255,255,255,254
710 DATA28,120,240,224,192,128,0,0
720 DATA255,255,231,231,255,127,31,15
730 DATA240,255,255,255,255,255,143,7
740 DATA255,255,255,255,255,255,127,28
750 DATA255,255,255,254,253,243,207,63
```

```
760 DATA252,216,176,96,192,192,128,0
770 DATA15,30,30,30,31,30,15,7
780 DATA3,1,16,96,128,3,7,0
790 DATA227,255,255,31,6,12,255,0
800 DATA254,252,248,224,0,0,0,0
810 DATA2,2,2,2,2,2,2,2
820 DATA255,66,36,24,24,36,66,255
830 REM >>>>> INITIALIZE <<<<<<
840 PX(1)=20:PX(2)=30:PX(3)=14:PX(4)=6
850 PY(1)=4:PY(2)=13:PY(3)=15:PY(4)=8
860 U1$=CHR$(96)+CHR$(97)+CHR$(98)+"      "
870 U2$=CHR$(99)+CHR$(100)+CHR$(101)+""
*
880 U3$=CHR$(32)+CHR$(102)+CHR$(103)+CHR$(104)+"      "
890 U4$=CHR$(32)+CHR$(105)+CHR$(106)+CHR$(107)+CHR$(108)+"      "
900 D1$=CHR$(124)+"      ":REM 5 SPACES
910 D2$=CHR$(109)+CHR$(110)+CHR$(111)+CHR$(112)+CHR$(113)+CHR$(114)
920 D3$=CHR$(115)+CHR$(116)+CHR$(117)+CHR$(118)+CHR$(119)+CHR$(32)
930 D4$=CHR$(120)+CHR$(121)+CHR$(122)+CHR$(123)+"      "
940 RETURN
950 REM >>> PRINT UPRIGHT PARROT <<<
960 X=PX(A):Y=PY(A):GOSUB510:PRINTU1$:Y=PY(A)+1:GOSUB510:PRINTU2$
970 Y=PY(A)+2:GOSUB510:PRINTU3$:Y=PY(A)+3:GOSUB510:PRINTU4$
980 RETURN
990 REM >>>> PRINT PARROT DOWN <<<<<
1000 X=PX(A):Y=PY(A):GOSUB510:PRINTD1$:Y=P(Y(A)+1:GOSUB510:PRINTD2$
1010 Y=PY(A)+2:GOSUB510:PRINTD3$:Y=PY(A)+3:GOSUB510:PRINTD4$:RETURN
1020 REM >>>>>> DRAW SCREEN <<<<<<<
1030 COLOR1,2,3:Y=0:FORX=0TO39:GOSUB510:PRINTCHR$(125);:NEXT
1040 COLOR1,3,5:FORT=1TO3:X=20:Y=T:GOSUB510:PRINTCHR$(124):NEXT
1050 COLOR1,8,5:FORT=1TO12:X=30:Y=T:GOSUB510:PRINTCHR$(124):NEXT
1060 COLOR1,4,5:FORT=1TO7:X=6:Y=T:GOSUB510:PRINTCHR$(124):NEXT
```

```

1070 COLOR1,7,6:FORT=1TO14:X=14:Y=T:GOSUB5
10:PRINTCHR$(124):NEXT
1080 PL$="" :FORT=1TO5:PL$=PL$+CHR$(125):NE
XT:COLOR1,2,3
1090 X=20:Y=8:GOSUB510:PRINTPL$:X=30:Y=17:
GOSUB510:PRINTPL$
1100 X=6:Y=12:GOSUB510:PRINTPL$:X=14:Y=19:
GOSUB510:PRINTPL$
1110 Y=20:FORX=0TO39:GOSUB510:PRINTCHR$(12
5)::NEXT
1120 COLOR1,9,4:X=17:Y=23:GOSUB510:PRINT"P
ARROT":RETURN
1130 REM >>>>>>> MAKE SOUND <<<<<<<
1140 IFNO(B)=1THEN A=1:COLOR1,3,5:GOSUB1000
:SOUND1,200,10:FORT=1TO500:NEXT
1150 IFNO(B)=2THEN A=2:COLOR1,8,5:GOSUB1000
:SOUND1,400,10:FORT=1TO500:NEXT
1160 IFNO(B)=3THEN A=3:COLOR1,4,5:GOSUB1000
:SOUND1,600,10:FORT=1TO500:NEXT
1170 IFNO(B)=4THEN A=4:COLOR1,6,5:GOSUB1000
:SOUND1,800,10:FORT=1TO500:NEXT
1180 GOSUB960
1190 RETURN
1200 X=0:Y=21:GOSUB510:FORZ=0TO30:PRINT" "
::NEXT
1210 Y=22:GOSUB510:FORZ=1TO30:PRINT" ";;:NE
XT:RETURN
1220 IFKS=49THEN D=1
1230 LE=0:RI=0:UP=0:DN=0:KS=PEEK(198)
1240 IFKS=10THEN LE=1
1250 IFKS=13THEN RI=1
1260 IFKS=43THEN UP=1
1270 IFKS=49THEN D=1
1280 RETURN
1290 POKE65298,196:POKE65299,209:STOP
1300 RESTORE1310:FORN=0TO53:READA:POKE828+
N,A:NEXT:RETURN
1310 DATA24,166,209,164,208,32,240,255,96,
169,0,133,210,169,12,133,211,32,60,3
1320 DATA164,209,240,8,169,40,32,102,3,136
,208,248,165,208,32,102,3,177,210,133
1330 DATA215,96,24,101,210,133,210,165,211
,105,0,133,211,96
1340 RESTORE1350:FORN=0TO30:READA:POKE882+
N,A:NEXT:RETURN

```

1350 DATA160,0,132,212,132,214,169,208,133
,213,169,56,133,215,162,3
1360 DATA177,212,145,214,209,208,249,230,2
13,230,215,202,16,242,96

19

Pick Man



An Arcade Gobble

This is the first of three arcade type games included in this book. It is a variation on a well known theme in which you guide your character around a maze eating up the dots as you go.

Life is never easy in the arcade world and so there are a couple of nasties out to eat up the dots before you do.

To make matters even worse they will kill you if they get you! The object of this variant is to pick up as many dots as possible before all the dots are gone, without being killed. Once all the dots are gone the next screen will be displayed.

The direction of the Pick Man is controlled using: cursor up for up,
* for down, S for right and A for left.

In the hands of a skilled and patient player, incredible high scores can be achieved.

```

10 REM <<<<< PICK MAN*ISSI/GREG >>>>>
20 GOSUB2020:PRINT"(CLR)":A=65298:POKEA,PE
EK(A)AND251
30 POKEA+1,(PEEK(A+1)AND3)OR56
40 POKE52,55:POKE54,55:POKE56,55:VOL8
50 GOSUB2070:SYS882
60 RESTORE70:FORN=0TO127:READA:POKE14848+N
,A:NEXT:GOTO260
70 DATA3,15,63,63,123,113,241,251
80 DATA192,240,252,252,222,142,143,223
90 DATA255,255,127,127,63,63,15,3
100 DATA255,255,254,254,252,252,240,192
110 DATA3,15,63,63,123,113,241,251
120 DATA192,240,252,252,222,142,143,223
130 DATA255,255,127,127,63,63,15,3
140 DATA255,255,254,254,252,252,240,192
150 DATA3,29,49,33,65,65,129,255
160 DATA255,129,65,65,33,49,29,3
170 DATA192,176,140,132,130,130,129,255
180 DATA255,129,130,130,132,140,176,192
190 DATA255,129,129,129,129,129,129,255
200 DATA3,0,0,0,0,0,0,0
210 DATA15,112,64,128,128,64,112,15
220 DATA240,14,2,1,1,2,14,240
230 YY$=LEFT$(YD$,Y):PRINTYY$,:PRINTTAB(X)
"GREG":STOP
240 SYS837:CH=PEEK(215):RETURN
250 SYS837:CM=PEEK(215):RETURN
260 HI=1000
270 REM >>>>>>> MAIN LOOP <<<<<<<<
280 REM
290 SC=0:CO=8:LI=3
300 GOSUB430:T$="(^*)(^A)":B$="(^B)(^C)"
310 T1$="(^D)(^E)":B1$="(^F)(^G)"
320 PRINT"(BLU)"
330 X=20:Y=15
340 X1=1:Y1=2:X2=35:Y2=2
350 GOSUB810
360 GOSUB1570
370 GOSUB810
380 IF (X1=XANDY1=Y) OR (X2=XANDY2=Y) THENDE=1
390 IF CO>257THEN1090
400 IF DE=1THEN1120
410 GOTO350
420 REM >>>>>>> SCREEN <<<<<<<<
```

```

430 COLOR4,1::COLOR0,1:PRINT"(HOME){OR}SC0
RE 00      LIVES ";LI
440 POKE208,28:POKE209,0:SYS828:PRINT"HIGH
"HI
450 COLOR1,6,5
460 W$=CHR$(108)+CHR$(108):X$=CHR$(104)+CH
R$(106):V$=CHR$(109)+CHR$(32)
470 Y$=CHR$(105)+CHR$(107):Z$=CHR$(110)+"{^L}{^L}{^L}{^L}{^L}{^L}{^L}"{^L}"+CHR$(111)
480 R$=V$+V$+V$:S$=R$+V$+V$
490 PRINTCHR$(104);:FORT=0TO37:PRINTCHR$(1
08);:NEXT:PRINTCHR$(106);
500 A$=CHR$(108)+S$+W$+S$+V$+V$+W$+S$+CHR$(
108)
510 PRINTA$;:Q$=A$
520 PRINTA$;
530 A$=CHR$(108)+V$+CHR$(110)+W$+W$+CHR$(1
06)+V$+Y$+V$+Z$+V$+Y$+V$
540 A$=A$+CHR$(104)+W$+W$+CHR$(111)+V$+CHR
$(108):PRINTA$;
550 A$=CHR$(108)+R$+W$+S$+S$+V$+W$+R$+CHR$(
108):PRINTA$;
560 MID$(A$,8,2)=Y$:MID$(A$,32,2)=Y$:PRINT
A$;
570 A$=W$+W$+CHR$(106)+R$+X$+V$+Z$+V$+X$+R
$+CHR$(104)+W$+W$:PRINTA$;
580 A$=W$+W$+CHR$(107)+R$+W$+S$+V$+V$+W$+R
$+CHR$(105)+W$+W$:PRINTA$;
590 A$=CHR$(108)+R$+X$+V$+W$+S$+V$+V$+W$+V
$+X$+R$+CHR$(108):PRINTA$;
600 A$=CHR$(107)+R$+W$+V$+Y$+V$+CHR$(104)+
W$+CHR$(107)+Y$
610 A$=A$+CHR$(105)+W$+CHR$(106)+V$+Y$+V$+
W$+R$+CHR$(105):PRINTA$;
620 A$=CHR$(32)+V$+X$+V$+W$+R$+W$+" "+W$+R$+W$+V$+X$+V$+CHR$(32):PRINTA$;
630 MID$(A$,4,2)=Y$:MID$(A$,36,2)=Y$
640 PRINTA$;
650 A$=CHR$(106)+R$+W$+V$+X$+V$+W$+" "+W$+V$+X$+V$+W$+R$+CHR$(104)
660 PRINTA$;
670 J$=A$:A$=CHR$(108)+MID$(J$,2,10)+W$+V$+
CHR$(105)+"{^L}{^L}{^L}{^L}{^L}{^L}{^L}"{^L}"+CHR$(107)
680 A$=A$+V$+W$+MID$(J$,30,10)+CHR$(108):P

```

```

RINTA$;
690 A$=W$+W$+CHR$(106)+V$+W$+V$+W$+S$+V$+V
$+W$+V$+W$+V$+CHR$(104)+W$+W$
700 PRINTA$;
710 MID$(A$,5,1)=CHR$(107):MID$(A$,36,1)=C
HR$(105):MID$(A$,8,2)=Y$
720 MID$(A$,32,2)=Y$:PRINTA$;
730 A$=CHR$(108)+S$+W$+V$+Z$+V$+W$+S$+CHR$
(108):PRINTA$; :A$=Q$:PRINTA$;
740 B$=CHR$(104)+W$+W$+CHR$(106):MID$(A$,4
,6)=B$:MID$(A$,32,6)=B$:PRINTA$;
750 B$=CHR$(105)+W$+W$+CHR$(107):MID$(A$,4
,6)=B$:MID$(A$,32,6)=B$
760 MID$(A$,16,10)=Z$:PRINTA$; Q$; Q$;
770 PRINTCHR$(105); :FORT=0TO37:PRINTCHR$(1
08); :NEXT:PRINTCHR$(107); :PRINT"(HOME)"
780 RETURN
790 REM >>>>>>> INITIALIZE <<<<<<<
800 REM >>>>>>> MOVEMENT <<<<<<<
810 GOSUB1750
820 IFLE=1THENGOSUB880
830 IFRI=1THENGOSUB920
840 IFUP=1THENGOSUB960
850 IFDN=1THENGOSUB990
860 GOSUB1690:RETURN
870 REM >>>>>>>> CHECKS <<<<<<<<
880 POKE208,X-1:POKE209,Y:GOSUB1840:IFCM<>
32THENRETURN
890 POKE208,X-1:POKE209,Y+1:GOSUB1840:IFCM
<>32THENRETURN
900 GOSUB1020:X=X-1:IFX=0THENX=37
910 RETURN
920 POKE208,X+2:POKE209,Y:GOSUB1840:IFCM<>
32THENRETURN
930 POKE208,X+2:POKE209,Y+1:GOSUB1840:IFCM
<>32THENRETURN
940 GOSUB1020:X=X+1:IFX=>38THENX=1
950 RETURN
960 POKE208,X:POKE209,Y-1:GOSUB1840:IFCM<>
32THENRETURN
970 POKE208,X+1:POKE209,Y-1:GOSUB1840:IFCM
<>32THENRETURN
980 GOSUB1020:Y=Y-1:RETURN
990 POKE208,X:POKE209,Y+2:GOSUB1840:IFCM<>
32THENRETURN

```

```
1000 POKE208,X+1:POKE209,Y+2:GOSUB1840:IFC
M<>32THENRETURN
1010 GOSUB1020:Y=Y+1:RETURN
1020 POKE208,X:POKE209,Y:SYS828:PRINT" ";
1030 POKE208,X:POKE209,Y+1:SYS828:PRINT"
":RETURN
1040 POKE208,X1:POKE209,Y1:SYS828:PRINT"
"
1050 POKE208,X1:POKE209,Y1+1:SYS828:PRINT"
"
1060 POKE208,X2:POKE209,Y2:SYS828:PRINT"
"
1070 POKE208,X2:POKE209,Y2+1:SYS828:PRINT"
":RETURN
1080 REM >>>>>>>> CLEAR <<<<<<<<
1090 SC=SC+1000:POKE208,6:POKE209,0:SYS828
:PRINTSC
1100 RESTORE2000:TU=12:GOSUB1820:CO=8:GOTO
300
1110 REM >>>>>>>> DEAD <<<<<<<<<
1120 C$=T$:D$=B$:DE=0:GOSUB1020:RESTORE198
0:TU=8:GOSUB1820
1130 GOSUB1040:COLOR1,9,4:LI=LI-1:IFLI=0TH
EN1920
1140 POKE208,19:POKE209,0:SYS828:PRINTLI:G
OT0330
1150 REM >>>>>> MOVE BADDIES <<<<<<
1160 POKE208,X1-1:POKE209,Y1:GOSUB240:IFCH
<>77ANDCH>71THENRETURN
1170 IFCH=77THENCO=CO+1
1180 POKE208,X1-1:POKE209,Y1+1:GOSUB240:IF
CH<>77ANDCH>71THENRETURN
1190 IFCH=77THENCO=CO+1
1200 GOSUB1040:X1=X1-1:GOSUB1710:RETURN
1210 POKE208,X2-1:POKE209,Y2:GOSUB240:IFCH
<>77ANDCH>71THENRETURN
1220 IFCH=77THENCO=CO+1
1230 POKE208,X2-1:POKE209,Y2+1:GOSUB240:IF
CH<>77ANDCH>71THENRETURN
1240 IFCH=77THENCO=CO+1
1250 GOSUB1040:X2=X2-1:GOSUB1710:RETURN
1260 POKE208,X1+2:POKE209,Y1:GOSUB240:IFCH
<>77ANDCH>71THENRETURN
1270 IFCH=77THENCO=CO+1
1280 POKE208,X1+2:POKE209,Y1+1:GOSUB240:IF
```

CH<>77ANDCH>71THENRETURN
1290 IFCH=77THENCO=CO+1
1300 GOSUB1040:X1=X1+1:GOSUB1710:RETURN
1310 POKE208,X2+2:POKE209,Y2:GOSUB240:IFCH
<>77ANDCH>71THENRETURN
1320 IFCH=77THENCO=CO+1
1330 POKE208,X2+2:POKE209,Y2+1:GOSUB240:IF
CH<>77ANDCH>71THENRETURN
1340 IFCH=77THENCO=CO+1
1350 GOSUB1040:X2=X2+1:GOSUB1710:RETURN
1360 POKE208,X1:POKE209,Y1-1:GOSUB240:IFCH
<>77ANDCH>71THENRETURN
1370 IFCH=77THENCO=CO+1
1380 POKE208,X1+1:POKE209,Y1-1:GOSUB240:IF
CH<>77ANDCH>71THENRETURN
1390 IFCH=77THENCO=CO+1
1400 GOSUB1040:Y1=Y1-1:GOSUB1710:RETURN
1410 POKE208,X2:POKE209,Y2-1:GOSUB240:IFCH
<>77ANDCH>71THENRETURN
1420 IFCH=77THENCO=CO+1
1430 POKE208,X2+1:POKE209,Y2-1:GOSUB240:IF
CH<>77ANDCH>71THENRETURN
1440 IFCH=77THENCO=CO+1
1450 GOSUB1040:Y2=Y2-1:GOSUB1710:RETURN
1460 POKE208,X1:POKE209,Y1+2:GOSUB240:IFCH
<>77ANDCH>71THENRETURN
1470 IFCH=77THENCO=CO+1
1480 POKE208,X1+1:POKE209,Y1+2:GOSUB240:IF
CH<>77ANDCH>71THENRETURN
1490 IFCH=77THENCO=CO+1
1500 GOSUB1040:Y1=Y1+1:GOSUB1710:RETURN
1510 POKE208,X2:POKE209,Y2+2:GOSUB240:IFCH
<>77ANDCH>71THENRETURN
1520 IFCH=77THENCO=CO+1
1530 POKE208,X2+1:POKE209,Y2+2:GOSUB240:IF
CH<>77ANDCH>71THENRETURN
1540 IFCH=77THENCO=CO+1
1550 GOSUB1040:Y2=Y2+1:GOSUB1710:RETURN
1560 RETURN
1570 N=RND(1):IFN>0.7THEN1620
1580 IFX>X1THEN GOSUB1260
1590 IFX<X1THEN GOSUB1160
1600 IFY>Y1THEN GOSUB1460
1610 IFY<Y1THEN GOSUB1360
1620 N=RND(1):IFN>0.6THENRETURN

```
1630 IFX>X2THENGOSUB1310
1640 IFX<X2THENGOSUB1210
1650 IFY>Y2THENGOSUB1510
1660 IFY<Y2THENGOSUB1410
1670 RETURN
1680 REM >>>>>>> PRINT MAN <<<<<<<<
1690 COLOR1,7,4:POKE208,X:POKE209,Y:SYS828
:PRINTT$;
1700 POKE208,X:POKE209,Y+1:SYS828:PRINTB$;
:RETURN
1710 COLOR1,3,2:POKE208,X1:POKE209,Y1:SYS8
28:PRINTT1$;
1720 POKE208,X1:POKE209,Y1+1:SYS828:PRINTB
1$;
1730 POKE208,X2:POKE209,Y2:SYS828:PRINTT1$;
;
1740 POKE208,X2:POKE209,Y2+1:SYS828:PRINTB
1$;:RETURN
1750 LE=0:RI=0:UP=0:DN=0:KS=PEEK(198)
1760 IFKS=10THENLE=1
1770 IFKS=13THENRI=1
1780 IFKS=43THENUP=1
1790 IFKS=49THENDN=1
1800 RETURN
1810 REM >>>>>>> PLAY TUNE <<<<<<<<
1820 FORN=1TOTU:READNO,DU:SOUND2,NO,DU:NEX
TN:RETURN
1830 REM>>>>>>>> DETECTION <<<<<<<<
1840 XX=PEEK(208):YY=PEEK(209):GOSUB250
1850 IFCM<>77THEN1880
1860 CM=32:SC=SC+10:SOUND2,120,7:CO=CO+1:P
OKE208,5:POKE209,0:SYS828:COLOR1,9,4
1870 PRINTSC:POKE208,XX:POKE209,YY
1880 IFCM=32THENRETURN
1890 IFCM>71THENRETURN
1900 DE=1:RETURN
1910 REM>>>>>>>> END GAME <<<<<<<<
1920 PRINT"(CLR)":POKE208,8:POKE209,3:SYS8
28:PRINT"(CYN)YOU HAVE BEEN GOBBLED !"
1930 POKE208,8:POKE209,21:SYS828:PRINT"(YE
L) PRESS 'Y' TO PLAY AGAIN"
1940 IFSC>HITHENHI=SC:POKE208,10:POKE209,9
:SYS828:PRINT"(PUR)A NEW HIGH GOBBLE !"
1950 GETA$:IFA$<>"Y"THEN1950
1960 GOTO280
```

```
1970 REM * * * DEAD TUNE * * *
1980 DATA284,20,284,20,284,20,358,75,319,2
0,319,20,319,20,379,75
1990 REM * * * SCREEN TUNE * * *
2000 DATA319,20,253,20,213,50,253,20,213,2
0,253,50
2010 DATA319,20,253,20,213,50,253,20,319,2
0,253,50
2020 RESTORE2030:FORN=0TO53:READA:POKE828+
N,A:NEXT
2030 DATA24,166,209,164,208,32,240,255,96,
169,0,133,210,169,12,133,211,32,60,3
2040 DATA164,209,240,8,169,40,32,102,3,136
,208,248,165,208,32,102,3,177,210,133
2050 DATA215,96,24,101,210,133,210,165,211
,105,0,133,211,96
2060 RETURN
2070 RESTORE2080:FORN=0TO30:READA:POKE882+
N,A:NEXT:RETURN
2080 DATA160,0,132,212,132,214,169,208,133
,213,169,56,133,215,162,3
2090 DATA177,212,145,214,200,208,249,230,2
13,230,215,202,16,242,96
```

20 **Skippy**



One Jump Ahead

'Once upon a time in Australia, which is a long way from Britain, there lived a little kangaroo. This little chap was called Skippy and he was very happy for many years.

'Skippy had his home by the banks of a fast flowing river on which logs floated. He used to enjoy sitting in his little hut looking out across the beautiful countryside, but one day things changed.

'The nasty humans came, and built a car expressway right next to the river. This meant that every day when Skippy and his friends returned from foraging for food, they had to cross the busy road, jump onto the logs to cross the river and then jump off as they passed their home. This is very dangerous and so they would like you, dear readers, to help.'

Using the cursor keys guide Skippy and his friends safely between the cars, onto the logs and into their homes. Your assistance to wildlife will be rewarded with points in relation to how well you guide each kangaroo!

```
213,230,215,202,16,242,96
1130 REM
1140 DATA 0,0,3,31,63,125,2,1
1150 DATA 127,132,4,255,255,223,32,192
1160 DATA 224,16,8,255,255,251,4,3
1170 DATA 0,0,0,248,252,188,64,128
1180 DATA 3,5,9,31,63,63,2,1
1190 DATA 242,18,19,240,255,222,33,192
1200 DATA 0,0,254,199,248,232,16,224
1210 DATA 0,0,0,255,0,0,0,0
1220 DATA 0,0,0,255,28,35,35,28
1230 DATA 0,0,0,255,225,17,16,224
1240 DATA 0,1,251,127,143,172,136,112
1250 DATA 57,234,252,206,237,112,49,0
1260 DATA 0,128,112,232,16,80,16,224
1270 DATA 255,132,114,41,161,85,173,87
1280 DATA 255,0,243,0,3,112,4,255
1290 DATA 255,48,192,14,128,48,0,255
1300 DATA 240,12,98,1,49,9,1,255
1310 DATA 0,127,127,127,127,127,127,127
1320 DATA 170,85,170,85,170,85,170,85
1330 DATA 255,255,255,255,255,255,255,255
1340 DATA 36,24,90,60,24,60,60,219
1350 DATA 36,24,24,60,90,189,126,136
1360 DATA 4,12,14,24,60,188,184,94
1370 DATA 8,24,28,24,60,61,58,220
1380 DATA 219,60,60,24,60,90,24,36
1390 DATA 36,126,189,90,60,24,24,36
1400 DATA 32,48,112,24,60,61,29,122
1410 DATA 16,24,56,24,60,188,92,59
```

```

760 COLOR1,9,4:CHAR1,0,2,BA$
770 COLOR1,7,3:PRINT RV$;RV$
780 COLOR1,9,4:FOR I=1 TO 5
790 PRINT B2$;:NEXT I
800 CHAR1,0,21,B2$:PRINT B2$;B3$
810 GOTO 500
820 REM ----- DEF SHAPES -----
830 RESTORE
840 FOR I=0 TO 30:READ J
850 POKE 882+I,J:NEXT I:SYS 882
860 FOR I=0 TO 223:READ J
870 POKE 14848+I,J:NEXT I
880 CR$=CHR$(96)+CHR$(97)+CHR$(98)+CHR$(99)
)
890 LY$=CHR$(100)+CHR$(101)+CHR$(102)+CHR$
(103)+CHR$(104)+CHR$(105)
900 BK$=CHR$(106)+CHR$(107)+CHR$(108)
910 LG$=CHR$(109)+CHR$(110)+CHR$(111)+CHR$
(110)+CHR$(111)+CHR$(112)
920 WZ$=CHR$(114)+CHR$(114)
930 W3$=W2$+CHR$(114)
940 S1$=CHR$(32):S2$=CHR$(32)+CHR$(32)
950 S3$=S1$+S2$:S4$=S2$+S2$
960 S8$=S4$+S4$:S0$=S4$+S8$
970 C1$=CR$+CR$+S8$+CR$+S4$+CR$+CR$+S3
$
980 C2$=LY$+S0$+LY$+S3$+LY$+LY$
990 C3$=BK$+BK$+S8$+BK$+BK$+BK$+BK$+BK$+S2
$+S3$
1000 L1$=LG$+W3$+W3$+LG$+W3$+W3$+LG$+W3$+W
3$+W3$
1010 L2$=LG$+W3$+W3$+LG$+W3$+W3$+LG$+W3$+W
3$+W3$
1020 FOR I=1 TO 100:RV$=RV$+CHR$(114):NEXT
1030 FOR I=1 TO 80:BA$=BA$+CHR$(115):NEXT
1040 FOR I=45 TO 75 STEP 10
1050 MID$(BA$,I,1)=CHR$(32):NEXT I
1060 FOR I=1 TO 40
1070 B2$=B2$+CHR$(113):B3$=B3$+CHR$(115)
1080 NEXT I
1090 RETURN
1100 REM ----- DATA -----
1110 DATA 160,0,132,212,132,214,169,208,13
3,213,169,56,133,215,162,3
1120 DATA 177,212,145,214,200,208,249,230,

```

```

390 REM ----- INPUT -----
400 TS=0:GET I$
410 IF I$="" THEN RETURN
420 TS=1 AND I$=CHR$(145) OR 3 AND I$=CHR$(
29):IF TS THEN RETURN
430 TS=5 AND I$=CHR$(17) OR 7 AND I$=CHR$(
157):RETURN
440 REM ----- MOVE CHARS -----
450 L2$=RIGHT$(L2$,1)+LEFT$(L2$,38)
460 L1$=RIGHT$(L1$,38)+LEFT$(L1$,1)
470 C3$=RIGHT$(C3$,1)+LEFT$(C3$,38)
480 C2$=RIGHT$(C2$,38)+LEFT$(C2$,1)
490 C1$=RIGHT$(C1$,1)+LEFT$(C1$,38)
500 COLOR1,7,3:CHAR1,0,5,L2$
510 CHAR1,0,7,L1$
520 COLOR1,3,2:CHAR1,0,15,C3$
530 COLOR1,8,6:CHAR1,0,17,C2$
540 COLOR1,7,3:CHAR1,0,19,C1$
550 COLOR1,2
560 RETURN
570 REM ----- MOVE KANGAROO -----
580 K=116+INT(TS/2)*2:M=TS
590 IF K AND 1 THEN SOUND 3,1000,4:ELSE SO
UND 2,200,4
600 IF (K AND 1)=0 THEN K=K+1:ELSE K=K-1
610 POKE 3072+KX+KY*40,CA:POKE 2048+KX+KY*
40,CC
620 KX=KX+(1 AND M=3 AND KX<38)-(1 AND M=7
AND KX>2)
630 KY=KY+(1 AND M=5 AND KY<23)-(1 AND M=1
)
640 CA=PEEK(3072+KX+KY*40):CC=PEEK(2048+KX
+KY*40)
650 CHAR1,KX,KY,CHR$(K)
660 RETURN
670 REM ----- SCORE ETC. -----
680 IF SC>HS THEN HS=SC
690 CHAR1,7,0,STR$(SC)
700 CHAR1,17,0,STR$(HS)
710 CHAR1,28,0,STR$(KA)
720 RETURN
730 REM ----- DRAW SCREEN -----
740 SCNCLR
750 COLOR1,2:CHAR1,0,0,"SCORE:      HIGH:
'ROOS:"

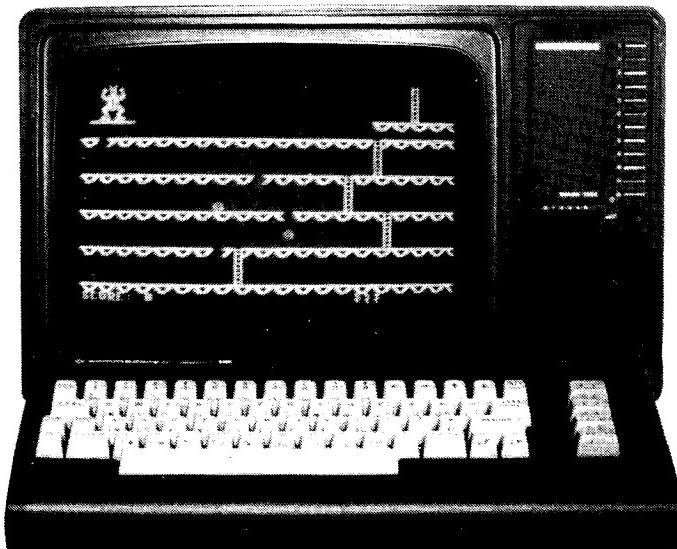
```

```
10 REM >>> SKIPPY <<<<<<<<
20 REM >>>BOOTS/GREG<<<<<<<
30 POKE52,55:POKE54,55:POKE56,55
40 POKE65298,192:POKE65299,(PEEK(65299) AN
D 3) OR 56
50 COLOR0,1:COLOR1,2:COLOR4,1:VOL 8
60 GOSUB 830
70 KA=3:SC=0
80 GOSUB 740:GOSUB 680:KH=0
90 KX=20:KY=21:K=116:CA=81:CC=72:M=0
100 CHAR1,KX,KY,CHR$(K)
110 REM ----- GAME LOOP -----
120 GOSUB 450:IF K AND 1 THEN GOSUB 590:GO
TO 120
130 CH=PEEK(3072+KX+KY*40)
140 IF CH>63 AND CH<77 OR CA>81 THEN 210
150 IF KY=3 THEN 310
160 GOSUB 400:IF TS THEN GOSUB 580
170 IF CA>67 AND CA<80 THEN KX=KX+(1 AND K
Y=5)-(1 AND KY=7)
180 IF KX=0 OR KX=39 THEN 210
190 GOTO 100
200 REM ----- LOSE LIFE -----
210 KA=KA-1
220 FOR I=800 TO 100 STEP -100
230 SOUND 2,I,1:NEXT I
240 IF KA THEN POKE 3072+KX+KY*40,CA:POKE
2048+KX+KY*40,CC:GOSUB 680:GOTO 90
250 SCNCLR
260 CHAR1,6,12," YOU BLEW IT YOU ROAD HOG
!"
270 CHAR1,4,14,"SPACE BAR FOR ANOTHER ATTE
MPT"
280 GETKEY I$:IF I$<>CHR$(32) THEN 280
290 GOTO 70
300 REM ----- MADE IT -----
310 FOR J= 1 TO 2
320 FOR I=100 TO 800 STEP 100
330 SOUND 2,I,1:NEXT I
340 FOR I=800 TO 100 STEP -100
350 SOUND 2,I,1:NEXTI,J
360 KH=KH+1:SC=SC+1:GOSUB 680
370 IF KH<4 THEN 90
380 CHAR1,6,12," WELL DONE YOU MADE IT OK
! ":GOTO 270
```

```
10 REM <<< KINKEY DONG <>ISSI/GREG >>>
20 A=65298:POKEA,PEEK(A)AND251:POKEA+1,(PE
EK(A+1)AND3)OR56:VOL8:PRINT"(CLR)"
30 POKE52,55:POKE54,55:POKE56,55:GOSUB2050
:GOSUB2090:SYS882:GOSUB420
40 GOSUB740
50 GOSUB710
60 IFLV=1THENGOSUB840
70 IFLV=2THENGOSUB910
80 IFLV=3THENGOSUB1010
90 GOSUB1200
100 X=1:Y=21
110 BX=1:BY=5:BD=1:BE=0
120 CX=37:CY=5:CD=-1:CE=0
130 IFLV=3THENBX=10:CX=27
140 GOSUB1640
150 GOSUB1770
160 GOSUB1680
170 POKE208,X:POKE209,Y:SYS828:PRINTOM$
180 GOSUB1290
190 IFY=0THEN280
200 POKE208,X:POKE209,Y:GOSUB430
210 IFGG=760RGG=32THEN230
220 GOTO300
230 POKE208,X:POKE209,Y+1:GOSUB430
240 IFGG=32THENY=Y+1
250 POKE208,BX:POKE209,BY:SYS828:PRINTOB$
260 POKE208,CX:POKE209,CY:SYS828:PRINTOC$
270 GOTO140
280 LV=LV+1:IFLV=4THENLV=1
290 SC=SC+(LV*50):POKE208,5:POKE209,23:SYS
828:PRINTSC:GOT060
300 SOUND2,300,75
310 LI=LI-1:IFLI=0THEN340
320 FORN=0TOLI
330 POKE208,28+N:POKE209,23:SYS828:PRINT"
":NEXT:GOT060
340 PRINT"(CLR)":POKE208,12:POKE209,8:SYS8
28
350 PRINT"(LT RED)TOUGH MONKEYS !"
360 PRINTTAB(11)^(GY 1)-----
370 POKE208,11:POKE209,12:SYS828:PRINT"(PU
R)YOU SCORED :"SC
380 PRINTTAB(11)^(BLU)(CUR DN)(CUR DN)PRES
S SPACE BAR"
```

21

Kinkey Dong



All Action Climax

This, the third arcade style game, is loosely inspired by a group of games in which the hero must jump over obstacles as he climbs up the ladder to get higher and higher through the screens.

The evil monster that appears on each screen has in fact abducted the hero's girlfriend and locked her up. He now spends all of his time rolling large barrels down to crush the hero.

There are different screen layouts and if you are successful then the game continues through from the first's screen to give higher scores.

Movement is by the 'S' for right, 'A' for left, 'Cursor up' for up, Asterisk for down. The little hero can be made to jump by pressing the space bar.


```

730 REM>>>>>>>> SET SCREEN <<<<<<
740 REM [ [ INITIALIZE ] ] ]
750 COLOR1,3,2
760 COLOR0,1:COLOR4,1:RETURN
770 REM >>>>>>> SCREEN <<<<<<<<
780 PRINT"(CLR)":COLOR1,3,3
790 POKE208,30:POKE209,4:SYS828:PRINTLEFT$(P$,9);
800 PO=34:N=0:GOSUB1250
810 COLOR1,3,3:FORN=6TO22STEP4
820 POKE208,0:POKE209,N:SYS828:PRINTP$;:NE
XTN:RETURN
830 REM * * * ONE * * *
840 GOSUB780:FORN=6TO18STEP4
850 PO=INT(RND(1)*36)+2
860 IFN<23THENPOKE208,PO:POKE209,N:SYS828:
PRINT" "
870 LP=RND(1)*36+2:IFLP=POTHEN870
880 IFN<23THENPO=LP:GOSUB1250
890 NEXT:RETURN
900 REM * * * TWO * * *
910 GOSUB780
920 FORN=11TO22
930 POKE208,6:POKE209,N:SYS828:PRINT"(*+){*
*+}{*+}{*+}{*+}{*+}{*+}":NEXT
940 POKE208,6:POKE209,17:SYS828:PRINT"
"
950 POKE208,6:POKE209,21:SYS828:PRINT"
"
960 POKE208,6:POKE209,13:SYS828:PRINT"
"
970 PO=8:N=6:GOSUB1250:PO=18:N=10:GOSUB125
0
980 PO=3:N=14:GOSUB1250:PO=18:N=18:GOSUB12
50
990 RETURN
1000 REM * * * THREE * * *
1010 PRINT"(CLR)":GOSUB440:COLOR1,3,3
1020 POKE208,0:POKE209,7:SYS828:PRINTGA$
1030 RESTORE1170:FORZZ=1TO6:READXP,YP,DU
1040 POKE208,XP:POKE209,YP:SYS828:PRINTLEF
T$(P$,DU):NEXT
1050 PO=17:N=6:GOSUB1250
1060 PO=20:N=6:GOSUB1250
1070 PO=13:N=10:GOSUB1250

```

```

1080 PO=24:N=10:GOSUB1250
1090 PO=15:N=14:GOSUB1250
1100 PO=22:N=14:GOSUB1250
1110 PO=6:N=18:GOSUB1250
1120 PO=31:N=18:GOSUB1250
1130 PO=34:N=0:GOSUB1250
1140 POKE208,11:POKE209,18:SYS828:PRINT" "
1150 POKE208,26:POKE209,18:SYS828:PRINT" "
1160 GOSUB1210:RETURN
1170 DATA30,4,10,9,6,20,7,10,24,5,14,28
1180 DATA3,18,32,0,22,39
1190 REM * * * TOP * * *
1200 GOSUB440
1210 POKE208,0:POKE209,23:SYS828:PRINT"SCO
RE:"SC;
1220 POKE208,28:POKE209,23:SYS828:FORN=1TO
LI:PRINTCHR$(111);:NEXT
1230 RETURN
1240 REM * * * LADDER * * *
1250 COLOR1,9,3:FORZ=0TO3
1260 POKE208,PO:POKE209,N+Z:SYS828:PRINTL$
1270 NEXT:RETURN
1280 REM >>>>>>>> MOVE <<<<<<<<<
1290 GOSUB1980
1300 IFLE=1ANDX>1THENX=X-1:M$=CHR$(112)
1310 IFRI=1ANDX<37THENX=X+1:M$=CHR$(111)
1320 IFUP=1THENM$=CHR$(113):GOTO1360
1330 IFDN=1THENM$=CHR$(113):GOTO1400
1340 IFFI=1THEN1420
1350 RETURN
1360 POKE208,X:POKE209,Y-1:GOSUB430:IFGG<>
76THEN1380
1370 Y=Y-1:GOTO1330
1380 POKE208,X:POKE209,Y+1:GOSUB430:IFGG=7
6THEN1370
1390 GOTO1330
1400 POKE208,X:POKE209,Y+1:GOSUB430:IFGG<>
76THEN1340
1410 Y=Y+1:GOTO1340
1420 IFM$=CHR$(112)THENXD=-1:ELSEXD=1
1430 IFX<10RX>37THENXD=0
1440 IFM$=CHR$(113)THENXD=0
1450 POKE208,BX:POKE209,BY:SYS828:PRINTOB$-
;
1460 POKE208,CX:POKE209,CY:SYS828:PRINTOC$-
;
```

```
1470 RESTORE1530:FORN=1T06
1480 READA
1490 Y=Y+A:GOSUB1540
1500 IFPL=1THENNN=11
1510 NEXTN
1520 GOSUB1680:RETURN
1530 DATA-1,-1,0,0,1,1
1540 X=X+XD:PL=0
1550 POKE208,X:POKE209,Y+1:GOSUB430:IFGG<>
32THENPL=1:RETURN
1560 IFX<1THENX=1
1570 IFX>38THENX=38
1580 GOSUB1640:GOSUB1680
1590 SOUND2,100,2
1600 POKE208,BX:POKE209,BY:SYS828:PRINTOB$;
;
1610 POKE208,CX:POKE209,CY:SYS828:PRINTOC$;
;
1620 GOSUB1770:POKE208,X:POKE209,Y:SYS828:
PRINTOM$;:RETURN
1630 REM [ [ [ MAN ] ] ]
1640 POKE208,X:POKE209,Y:GOSUB430
1650 OM$=CHR$(GG+32):IFGG=32THENOM$=CHR$(3
2)
1660 PRINTM$;:RETURN
1670 REM [ [ [ BARRELS ] ] ]
1680 POKE208,BX:POKE209,BY:GOSUB430
1690 OB$=CHR$(GG+32):IFGG=32THENOB$=CHR$(3
2)
1700 PRINTB$;:IFGG=790RGG=800RGG=81THEN300
1710 POKE208,CX:POKE209,CY:GOSUB430
1720 OC$=CHR$(GG+32):IFGG=32THENOC$=CHR$(3
2)
1730 IFGG=77THENOC$=" "
1740 PRINTB$;:IFGG=790RGG=800RGG=81THEN300
1750 RETURN
1760 REM>>>>>>>> BARREL <<<<<<<<<<
1770 A=INT(RND(1)*10)
1780 POKE208,BX:POKE209,BY+1:GOSUB430
1790 IFGG=320RGG=76ANDA>5THENBE=1
1800 IFGG=780RGG=82THENBE=0
1810 BY=BY+BE:IFBE=0THENBX=BX+BD
1820 POKE208,CX:POKE209,CY+1:GOSUB430
1830 IFGG=320RGG=76ANDA>5THENCE=1
1840 IFGG=780RGG=82THENCE=0
```

```
1850 CY=CY+CE: IF CE=0 THEN CX=CX+CD
1860 IF BX<1 THEN BD=1
1870 IF BX>37 THEN BD=-1
1880 IF CX<1 THEN CD=1
1890 IF CX>37 THEN CD=-1
1900 IF (BX<10 OR BX>37) AND BY=21 THEN BX=1: BD=1:
BY=5: GOSUB 1930
1910 IF (CX<10 OR CX>37) AND CY=21 THEN CX=37: CD=-
1: CY=5: GOSUB 1950
1920 RETURN
1930 IF LV=3 THEN BX=10
1940 RETURN
1950 IF LV=3 THEN CX=27
1960 RETURN
1970 RETURN
1980 LE=0: RI=0: UP=0: DN=0: FI=0: KS=PEEK(198)
1990 IF KS=10 THEN LE=1
2000 IF KS=13 THEN RI=1
2010 IF KS=43 THEN UP=1
2020 IF KS=49 THEN DN=1
2030 IF KS=60 THEN FI=1
2040 RETURN
2050 RESTORE 2060: FOR N=0 TO 53: READ A: POKE 828+N, A: NEXT: RETURN
2060 DATA 24, 166, 209, 164, 208, 32, 240, 255, 96,
169, 0, 133, 210, 169, 12, 133, 211, 32, 60, 3
2070 DATA 164, 209, 240, 8, 169, 40, 32, 102, 3, 136
, 208, 248, 165, 208, 32, 102, 3, 177, 210, 133
2080 DATA 215, 96, 24, 101, 210, 133, 210, 165, 211
, 105, 0, 133, 211, 96
2090 RESTORE 2100: FOR N=0 TO 30: READ A: POKE 882+N, A: NEXT: RETURN
2100 DATA 160, 0, 132, 212, 132, 214, 169, 208, 133
, 213, 169, 56, 133, 215, 162, 3
2110 DATA 177, 212, 145, 214, 200, 208, 249, 230, 2
13, 230, 215, 202, 16, 242, 96
```

22

Word Splash



Spell or Swim

This game is a more humane, more fun version of the well known hangman game. If the player cannot guess the word in time, then all the wrong letters push a character into the water.

Words can be up to eight letters long but should not contain hyphens. The listing has about 60 words already included. More words may be added or the existing ones changed.

If a letter is guessed, it appears at each correct position in the word. If it is not right then the poor victim is pushed one space along. Should the word be completed in time then the little figure pushes all the wrong letters back and a new mystery word is offered.

The word is not shown if it is not guessed and since words are randomly selected it may be possible to have the same word again later.

The program is very entertaining, educational and can be surprisingly addictive!

```
10 REM<<WORD SPLASH * ISSI/GREG/JIM>>
20 POKE52,55:POKE54,55:POKE56,55:VOL8
30 PRINT"(CLR)":GOSUB1180:GOSUB1200:SYS8
40 A=65298:POKEA,PEEK(A)AND251:POKEA+1,(PE
EK(A+1)AND3)OR56
50 GOSUB690
60 W=INT(RND(1)*60)+1
70 W$=D$(W)
80 GOSUB500
90 GOSUB890
100 FL=0
110 IFT$=W$THEN270
120 IFLEN(G$)=10THEN150
130 GOTO90
140 REM * * * LOSE * * *
150 GOSUB750
160 RESTORE210
170 FORN=1TO10
180 READA,B
190 SOUND1,A,B
200 NEXTN
210 DATA253,20,284,20,319,50,284,20
220 DATA319,20,358,50,319,20,358,20
230 DATA379,50,358,50
240 PRINT"(CLR)":POKE208,15:POKE209,6:SYS8
28:PRINT"(CYN)TOUGH LUCK !"
250 GOTO450
260 REM * * * WIN * * *
270 RESTORE320
280 FORN=1TO16
290 READA,B
300 SOUND1,A,B
310 NEXTN
320 DATA319,20,284,20,253,20,239,80
330 DATA319,20,284,20,253,20,239,80
340 DATA319,20,284,20,253,20,239,40
350 DATA284,40,358,40,284,40,319,80
360 IFG$=""THEN430
370 FORN=LEN(G$)TO1STEP-1
380 G$=RIGHT$(G$,N-1)
390 GOSUB1010:PRINT" "
400 SOUND1,N*20,2
410 FORP=1TO500
420 NEXTP,N
430 PRINT"(CLR)"
```

```
440 POKE208,12:POKE209,5:SYS828:PRINT"(PUR
)CONGRATULATIONS !"
450 POKE208,13:POKE209,18:SYS828:PRINT"(GY
33PRESS SPACE BAR"
460 KS=PEEK(198)
470 IFKS<>60THEN460
480 GOTO60
490 REM >>>>>>>>> SCREEN <<<<<<<<
500 COLOR0,1:COLOR4,1:PRINT"(CLR)"
510 POKE208,15:POKE209,2:SYS828:PRINT"(PUR
)WORD SPLASH"
520 POKE208,14:POKE209,3:SYS828:PRINT"(YEL
3-----"
530 A$="GY 1}{*+}{*+}{RVS ON}{RED}
{**}{RVS OFF}"
540 POKE208,4:POKE209,11:SYS828:PRINTA$
550 R$="**}{RVS ON}{**}{RVS OFF}":POKE208
,16:POKE209,12:SYS828:PRINTR$
560 P$="GRN}{RVS ON}{BLU}
{GRN}{RVS OFF}"
570 POKE208,16:POKE209,13:SYS828:PRINT"(RV
S ON}{GRN}{**}{RVS OFF}"
580 POKE208,32:POKE209,13:SYS828:PRINT"(RV
S ON){^$}{RVS OFF}"
590 FORN=14TO21:POKE208,16:POKE209,N:SYS82
8:PRINTP$:NEXT
600 POKE208,16:POKE209,22:SYS828:PRINT"(RV
S ON}{OR} {RVS OFF}"
610 POKE208,4:POKE209,13:SYS828:PRINT"(WHT
)WORD :-
620 POKE208,7:POKE209,17:SYS828:T$=""":G$="
"
630 FORN=1TOLEN(W$)
640 PRINT"(BLU)-";T$=T$+" "
650 NEXTN
660 POKE208,7:POKE209,20:SYS828:PRINT"(LT
RED}GUESS ?"
670 RETURN
680 REM>>>>>>>> INITIALIZE <<<<<<<
690 DIMD$(60):RESTORE1050
700 FORN=1TO60:READD$(N)
710 NEXT
720 RESTORE1130:FORN=0TO39:READA:POKE14848
+N,A:NEXT
730 RETURN
```

740 REM>>>>>>> SPLASH <<<<<<<<<
750 POKE208,16:POKE209,10:SYS828
760 M\$=CHR\$(A)
770 PRINT" "
780 X=16:Y=10:FORZ=1TO3:POKE208,X:POKE209,
Y:SYS828:PRINT" "
790 X=X+1:Y=Y+1:POKE208,X:POKE209,Y:SYS828
:PRINTM\$
800 SOUND1,700,5:FORT=1TO60:NEXT
810 POKE208,X:POKE209,Y:SYS828:PRINTM\$
820 NEXT
830 COLOR1,7,5:SH\$=CHR\$(98)+CHR\$(99)+CHR\$(
100)
840 POKE208,X-1:POKE209,Y:SYS828:PRINTSH\$
850 SOUND3,900,30:FORT=1TO220:NEXT
860 POKE208,X-1:POKE209,Y:SYS828:PRINT"
-
870 FORT=1TO30:NEXT:RETURN
880 REM>>>>>>> INPUT <<<<<<<<<
890 GETA\$
900 IF A\$<"A" OR A\$>"Z" THEN 890
910 SOUND2,200,10
920 FL=0
930 FORN=1TOLEN(W\$)
940 IF MID\$(W\$,N,1)<>A\$ THEN 960
950 POKE208,6+N:POKE209,16:SYS828:PRINTA\$:
MID\$(T\$,N,1)=A\$:FL=1
960 NEXTN
970 IFFL=0 THEN 1000
980 SOUND1,650,20
990 RETURN
1000 G\$=A\$+G\$
1010 POKE208,6:POKE209,10:SYS828
1020 IF LEN(G\$)/2=INT(LEN(G\$)/2) THEN A=96:EL
SEA=97
1030 PRINTG\$;CHR\$(A);CHR\$(32)
1040 RETURN
1050 DATA ACUTE, AGILE, BROKE, CANDY, COUGH, DRY
LY, GNOME, MISTY, NYMPH, PYGMY
1060 DATA BARLEY, BODKIN, CEPTIC, DEVOUR, EIGHT
Y, EXOTIC, FLINCH, GLORIA, HYMN, LYRICS
1070 DATA METHYL, SPYING, SYRUP, YOGHURT, VISUA
L, WOBBLE, ANCHOVY, RELAX, RHYTHM, RHUBARB
1080 DATA SKETCH, FASCIST, BORING, FATIGUE, FRO
WN, TRUANT, LEMMING, ZINC, GLAMOUR

```
1090 DATAHYDRATED
1100 DATAKETCHUP,MYSTIC,NOISY,OCTOPUS,SQUI
D,RAMBLER,SATCHEL,SEXTANT,YULETIDE
1110 DATASWINDLE,PLAYER
1120 DATADOCKER,DESTINY,FUTURE,LENTIL,FORE
IGN,LOGICAL,WINTER,THIRST,NAIVE
1130 DATA56,56,18,124,144,40,36,34
1140 DATA56,56,144,124,18,40,72,136
1150 DATA1,16,72,36,2,96,24,7
1160 DATA32,161,164,9,34,52,123,124
1170 DATA130,16,36,8,51,192,12
1180 RESTORE1190:FORN=0TO8:READA:POKE828+N
,A:NEXT:RETURN
1190 DATA24,166,209,164,208,32,240,255,96
1200 RESTORE1210:FORN=0TO30:READA:POKE892+N
,A:NEXT:RETURN
1210 DATA160,0,132,212,132,214,169,208,133
,213,169,56,133,215,162,3
1220 DATA177,212,145,214,200,208,249,230,2
13,230,215,202,16,242,96
```

23

Sum Fun



Ten out of Ten

This single program can offer questions for 'primary', 'junior' and 'senior' levels (that is everyone). The question can be of any one type, selected from plus, minus, multiply and divide. The questions and answers are drawn to a blackboard to help make the presentation attractive.

Selecting the primary level will set up problems which involve counting two sets of characters to produce a total. The junior level provides options for the four types of arithmetic and produces questions which have whole number answers. It should be possible for most adults and older children to work these out mentally.

The senior level would require a very strong mental arithmetic ability and is designed to set problems which can be worked out using pencil and paper or even a calculator.

The answers are required to be typed in and then sent to the computer using the 'RETURN' key. This means that mistyped answers can be corrected if spotted before the 'RETURN' key is

pressed. The program will allow up to two wrong trys before revealing the correct answer.

At the end of ten questions the computer will print up a 'report' showing the number of correct answers and a percentage score.

```

10 REM <<<<< SUM FUN <> ISSI/JIM>>>
20 REM
30 A=RND(-T1):VOL 8:GOSUB 70
40 CHAR ,12,7,"1>PRIMARY",1:CHAR ,12,9,"2>
JUNIOR",1:CHAR ,12,11,"3>SENIOR",1
50 GETKEY A$:IF A$<"1" OR A$>"3" THEN GOTO
50
60 SOUND 1,400,10:FOR N=1 TO 200:GET B$:NE
XT N:LEV=VAL(A$):GOTO 410
70 COLOR 0,15,0:COLOR 4,15,2:COLOR 1,10,1:
GRAPHIC 1,1
80 DRAW 1,168,10 TO 208,175 TO 200,175 TO
164,20 TO 120,175 TO 112,175
90 DRAW TO 160,10 TO 168,10:PAINT 1,164,15
100 GOTO 470
110 G$=""
120 CHAR ,12,13,G$,1:GETKEY A$:SOUND 1,500
,10
130 IF (A$)= "0" AND A$<="9")OR A$="--"OR (A
$=CHR$(20)AND LEN(G$)>0) THEN 170
140 IF G$="" OR A$<>CHR$(13) THEN 120
150 IF VAL(G$)=ANS THEN RIT=1:ELSE RIT=0
160 CHAR ,12,13," ",1:RETUR
N
170 CHAR ,12,13,G$:IF A$=CHR$(20) THEN G$=
LEFT$(G$,LEN(G$)-1):A$=""
180 G$=G$+A$:GOTO 120
190 FOR G=1 TO 2:GOSUB 110:IF RIT=1 THEN 2
20
200 SOUND 1,300,15:SOUND 1,500,25:NEXT G
210 GOTO 470
220 SOUND 1,300,10:SOUND 1,400,10:SOUND 1,
500,10:G=2:NEXT G:TT=TT+1:GOTO 470
230 GOSUB 470:CHAR ,14,7,"1> '+'",1:CHAR
,14,9,"2> '-'",1
240 CHAR ,14,11,"3> '*'",1:CHAR ,14,13,"4
> '/'",1:DEF FNI(W)=A+B:S$="+"
250 GETKEY A$:Z=VAL(A$):IF Z<1 OR Z>4 THEN
250
260 SOUND 1,600,10

```

```
270 IF Z=2 THEN DEF FNI(W)=A-B:S$="-"
280 IF Z=3 THEN DEF FNI(W)=A*B:S$="*"
290 IF Z=4 THEN DEF FNI(W)=A/B:S$="/" :DIV=
1
300 RETURN
310 LIM=8:DIV=0:IF LEV=2 THEN LIM=12:GOSUB
230
320 IF LEV=3 THEN LIM=25:GOSUB 230
330 TT=0:GOTO 470
340 A=INT(RND(1)*LIM)+1:B=INT(RND(1)*LIM)+1
350 IF DIV=1 THEN A=A*B
360 GOSUB 470:N$=STR$(P):CHAR ,10,6,N$+">",
1
370 IF LEV=1 THEN 390
380 Q$=STR$(A)+" "+S$+STR$(B):CHAR ,12,8,Q$,
1:ANS=FNI(1):RETURN
390 A$="":FOR N=1 TO A:A$=A$+"{^S}":NEXT N
B$="":FOR N=1 TO B:B$=B$+"{^X}":NEXT N
400 CHAR ,12,8,A$+" ",1:CHAR ,12,10,B$+"",
1:ANS=A+B:RETURN
410 GOSUB 310:FOR P=1 TO 10:GOSUB 340:GOSU
B 190:NEXT P:GOSUB 470
420 CHAR ,17,6,"REPORT.",1:CHAR ,16,7,"---
---",1
430 W$=STR$(TT)+" WERE CORRECT.":CHAR ,12,
10,W$,1
440 W$="THAT'S "+STR$(TT/10*100)+" %":CHAR
,14,12,W$,1
450 CHAR ,10,15,"PRESS SPACE TO PLAY.",1:GETKEY
A$:IF A$<>" " THEN 450
460 GOSUB 470:GOTO 40
470 W$="":FOR N=1 TO 20:W$=W$+":NEXT N:C
OLOR 1,1
480 FOR N=5 TO 15:CHAR 1,10,N,W$,1:NEXT N:
RETURN
```

24

Today England



Tomorrow the World

This is a geographic quiz based upon the map of Great Britain. The game is very straightforward and consists of questions which relate to key towns or cities, and the county to which they belong. Options are given to guess the county or the town. A point is plotted on the map to indicate the location of the town or city.

The central core of the program could be developed to produce either a more detailed map or, as the heading suggests, the world could be plotted next!

Another way to develop this program would be to extend the data to include more or all counties and the county towns. In fact you could find that the research of the facts to put into the listing is even more educational than the program itself! Due to memory restrictions the program must be entered and saved as two separate programs. The first will then load the second when it is run. If you are using disk then the device number '8' will have to be added to the load instruction.

```
10 REM>>>>TODAY ENGLAND<<<<<
20 REM>>>>GREG<<<<<<<<<
30 COLOR0,1:COLOR4,1:GRAPHIC1:SCNCLR
40 GOSUB70
50 POKE45,207:POKE46,21:POKE47,207:POKE48,
21:POKE49,207:POKE50,21:CLR
60 LOAD"GAME":RUN
70 COLOR1,2,7:DRAW1,96,14:FORN=1TO58
80 READX,Y:Y=199-INT(Y/2):X=X/2:DRAWTOX,Y:
NEXT
90 DRAW1,71,78:FORN=1TO16:READX,Y:Y=199-IN
T(Y/2):X=X/2:DRAWTOX,Y:NEXT
100 DRAW1,96,108:FORN=1TO3:READX,Y:Y=199-(Y/2):X=X/2:DRAWTOX,Y:NEXT
110 DRAW1,103,120:FORN=1TO3:READX,Y:Y=199-(Y/2):X=X/2:DRAWTOX,Y:NEXT
120 DRAW1,143,171:FORN=1TO4:READX,Y:Y=199-(Y/2):X=X/2:DRAWTOX,Y:NEXT:RETURN
130 DATA228,377,234,359,210,329,234,341,25
8,335,258,305,228,287
140 DATA252,275,228,269,252,263,270,245,27
6,215,306,203,318,173,294,179,324
150 DATA149,318,137,354,143,360,113,324,83
,354,77,330,53,282,61,253,47,234,53
160 DATA216,35,174,23,165,24,210,71,246,71
,258,95,234,83,204,95,198,107,216
170 DATA113,216,137,198,131,216,155,252,15
5,240,173,246,191,234,191,222,209
180 DATA234,227,204,215,192,227,204,245,19
8,263,180,239,168,245,180,287,162
190 DATA281,180,317,168,341,192,347,180,36
5,198,359,192,371
200 DATA172,229,184,199,172,187,160,187,17
2,139,160,109,118,103,82,85
210 DATA58,80,80,130,100,157,70,163,82,199
,110,200,120,240,142,241
220 DATA204,181,210,199,192,181
230 DATA206,151,210,160,206,157
240 DATA280,49,292,43,298,49,286,55
250 SC=0
260 GRAPHIC0:SCNCLR:COLOR1,5,4:CHAR1,14,1,
*TODAY ENGLAND*
270 COLOR1,8,5:CHAR1,12,3,"TOMORROW THE WO
RLD":COLOR1,7,5
280 CHAR1,1,6,"CHOOSE OPTION:":COLOR1,9,5:
```

```

CHAR1,11,8,"1...GUESS COUNTIES"
290 CHAR1,11,10,"2...GUESS CITIES":SC$="SC
ORE = "+STR$(SC):CHAR1,11,12,SC$
300 SC=0
310 GETKEYA$:O=VAL(A$):IFO<10R0>2THEN310
320 GRAPHIC1:RESTORE490
330 FORC=1TO16:COLOR1,2,7:READN$,X,Y,CO$
340 X=X/2:Y=199-(Y/2):DRAW1,X,Y:TR=3
350 IFO=1THENCHAR1,20,1,"CITY IS ?":CHAR1,
20,3,N$
360 IFO=2THENCHAR1,20,1,"COUNTY IS ?":CHAR
1,20,3,CO$
370 IFO=1THENCHAR1,20,5,"COUNTY IS ?":ELSE
CHAR1,20,5,"CITY IS ?"
380 H=20:V=7:GOSUB590:IFO=1ANDDT$=CO$THEN4
30
390 IFO=2ANDDT$=N$THEN430
400 TR=TR-1:TR$=STR$(TR)+" TRIES LEFT":CHA
R1,20,9,TR$
410 FORT=1TO800:NEXT:CHAR0,20,9,"00 TRIES
LEFT"
420 IFTR=0THEN570:ELSEGOTO350
430 GOSUB480
440 CHAR1,20,9,"CORRECT":FORT=1TO800:NEXT:
SC=SC+TR
450 DRAW0,X,Y
460 GOSUB480:NEXT
470 FORT=1TO800:NEXT:GOT0260
480 DT$=""":FORT=1TO19:DT$=DT$+" ":NEXT:FOR
T=1TO10:CHAR1,20,T,DT$":NEXT:RETURN
490 DATA MANCHESTER,258,171,GTR.MANCHESTER,
CHESTER,252,149,CHESHIRE
500 DATA LEEDS,282,175,YORKSHIRE,HULL,308,1
81,HUMBERSIDE
510 DATA LIVERPOOL,250,163,MERSEYSIDE,BIRMI
NGHAM,274,119,MIDLANDS
520 DATA LONDON,318,85,GTR.LONDON,BRISTOL,2
60,83,AVON
530 DATA CARDIFF,244,95,GLAMORGAN,CAMBRIDGE
,322,115,CAMBRIDGESHIRE
540 DATA GLASGOW,216,261,STRATHCLYDE,EDINBU
RGH,238,261,LOTHIAN
550 DATA ABERDEEN,250,315,GRAMPIAN,NEWCASTL
E,266,223,NORTHUMBERLAND
560 DATA BELFAST,170,200,ARMAGH,DUBLIN,160,

```

```
160,DUBLIN
570 GOSUB480:CHAR1,20,1,"CITY IS":CHAR1,20
,3,N$:CHAR1,20,5,"COUNTY IS"
580 CHAR1,20,7,CO$:FORT=1TO1800:NEXT:GOTO4
50
590 DT$="""
600 GETKEYA$
610 IF A$=CHR$(13) THEN 640
620 IF A$=CHR$(20) THEN DT$=LEFT$(DT$,LEN(DT$)
)-1):A$="""
630 DT$=DT$+A$:D1#=DT$+" ":CHAR1,H,V,D1$:G
0TO600
640 RETURN
```

25

Character Builder



The Create Your Own Monster Kit

Character building can be as much fun as playing a game. This 'utility' enables 32 characters to be defined easily.

The built-in characters from 96 to 127 are first displayed and then the opportunity is given to adapt them, or create completely new ones. The definition values can be noted down and then used in your own programs. Complete sets can be saved on tape to produce a library of useful characters.

Although many of the programs in this book have had characters designed for them, others use standard characters. With this character definer you can dramatically improve the appearance of these games. You may even wish to change all the ones in the book!

Character builder reference guide

After loading, the grid starts with the first character picked up. This

may then be Edited, Cleared, or any of the following can be selected by pressing the first letter of the option.

Clear. This clears the grid completely.

Pick up. Asks for character number, then as the last digit is pressed that character will be transferred to the grid. It can then be Edited, Changed, etc. Note: three characters must be entered, therefore 2 would be ØØ2.

Edit. Places the flashing cursor at the top left of the grid. It can then be moved around using A, S, * and cursor up. Space bar will set or unset a point. When finished Q is pressed.

Press 'T' for tape save.

Press 'L' to load data from tape.

```

10 REM << CHAR BUILDER * ISSI/JIM/GREG >>
20 POKE52,55:POKE54,55:POKE56,55:GOSUB1010
:SYS882
30 POKE65298,192:POKE65299,(PEEK(65299)AND
3)OR56
40 DIMF(8),D(32,8)
50 CH=1
60 SCNCLR:GOSUB710
70 GOSUB580
80 GOSUB770
90 REM >>>>>>>> MAIN LOOP <<<<<<<<
100 GETKEYA$
110 IFA$="C":THENGOSUB170
120 IFA$="P":THENGOSUB210
130 IFA$="E":THENGOSUB230
140 IFA$="T":THENGOTO450
150 IFA$="L":THENGOTO520
160 GOTO100
170 A$="           ":REM 8 SPACES
180 FORY=2TO9:CHAR1,26,Y,A$:B$=STR$(0)+"
":CHAR1,35,Y,B$:NEXT
190 CR$="      ":GOSUB680:RETURN
200 REM >>>>>>>> PICK UP <<<<<<<<<
210 GOSUB170:GOSUB860:GOSUB770:GOSUB670:RE
TURN
220 REM >>>>>>>>>> EDIT <<<<<<<<<<
230 H=26:V=2:COLOR1,2,7
240 LO=3072+V*40+H:AB=PEEK(LO):POKELO,AB+1
28:FORT=1TO30:NEXT
250 POKELO,AB
260 GOSUB870

```

```

270 IFLE=1ANDH>26THENH=H-1
280 IFRI=1ANDH<33THENH=H+1
290 IFUP=1ANDV>2THENV=V-1
300 IFDN=1ANDV<9THENV=V+1
310 IFQ=1THENA$="" :RETURN
320 IFFI=0THEN390
330 G=H-25:X=V-1
340 IFAB=32THENDD=42:W=BI(G):ELSEDD=32:W=0
350 POKELO, DD:J=D(CH,X):IFW=0THEN370
360 J=JORW:D(CH,X)=J:GOTO380
370 C=BI(G):W=(NOTC)+1:K=D(CH,X):J=K+W:D(C
H,X)=J
380 GOSUB410:SS=14848+((CH-1)*8)+X-1:POKES
S,J
390 GOTO240
400 REM >>>>>>>>> VIEW <<<<<<<<<
410 BB=X:X=D(CH,BB):GOSUB820:CHAR1,26,V,X$
420 P$="" :XX$=STR$(X):CHAR1,36,V,P$:CHA
R1,35,V,XX$
430 X=BB:RETURN
440 REM >>>>>>>>> SAVE <<<<<<<<
450 RESTORE490:FORT=0TO31:READA:POKE15360+
T,A:NEXT
460 CHAR1,0,21," "
470 SYS15360
480 GOTO60
490 DATA169,1,162,1,160,255,32,186,255,169
,0,32,189,255,169,0,133,208,169,58
500 DATA133,209,162,0,160,59,169,208,32,21
6,255,96
510 REM >>>>>>>> LOAD <<<<<<<<<
520 RESTORE560:FORT=0TO23:READA:POKE15392+
T,A:NEXT
530 CHAR1,0,21," "
540 SYS15392
550 GOTO60
560 DATA169,1,162,1,160,255,32,186,255,169
,0,32,189,255,169,0,162,255,160
570 DATA255,32,213,255,96
580 COLOR0,1:COLOR4,1:COLOR1,8,5:SCNCLR
590 CHAR1,0,0,"NO.    CHAR      NO.    CHAR"
600 FORX=0TO23:CHAR1,X,1,"-":NEXT
610 FORY=3TO18:D$=STR$(Y+93):CHAR1,0,Y,D$:
CHAR1,7,Y,CHR$(Y+93):NEXT
620 FORY=3TO18:D$=STR$(Y+109):CHAR1,13,Y,D

```

```

$ : CHAR1, Z1, Y, CHR$(Y+109) : NEXT
630 COLOR1, 7, 4: A$ = " {RVS ON}           {RVS
OFF}" : CHAR1, 25, 1, A$, 1: FORT=1TO8
640 B$ = " {RVS ON} {RVS OFF}           {RVS ON}
{RVS OFF}" : CHAR1, 25, T+1, B$: NEXT
650 CHAR1, 25, 10, A$, 1
660 COLOR1, 6, 5: CHAR1, 0, 21, "CHARACTER DISPLAYED:"
670 COLOR1, 6, 5: CR$=STR$(CH+95) + " "
680 CHAR1, 20, 21, CR$
690 RETURN
700 REM >>> INITIALIZE VARIABLES <<<
710 AD=14848
720 FORN=1TO32: FORM=1TO8: D(N,M)=PEEK(AD): A
D=AD+1:NEXTM,N
730 RESTORE740: FORT=1TO8: READBI(T): NEXT
740 DATA128, 64, 32, 16, 8, 4, 2, 1
750 RETURN
760 REM >>>>>> DRAW CHAR <<<<<<<<
770 Y=2: BB=X: COLOR1, 2, 7
780 FORN=1TO8: X=D(CH, N): GOSUB820
790 CHAR1, 26, Y, X$: XX$=STR$(X): P$= " " : CHA
R1, 36, Y, P$: CHAR1, 35, Y, XX$: Y=Y+1
800 NEXT
810 X=BB: RETURN
820 X$= "00000000": U=128
830 FORM=1TO8: Z=UANDX: IFZ<>0THENB$="*": ELS
EB$= " "
840 U=U/2: MID$(X$, M, 1)=B$: NEXT: RETURN
850 REM >>>>>> INPUT CHAR <<<<<<<
860 COLOR1, 6, 5: CHAR1, 24, 13, "CHARACTER ?": X
=35: Y=13: GOSUB950: RETURN
870 Q=0: LE=0: RI=0: UP=0: DN=0: FI=0: KS=PEEK(1
98)
880 IF KS=10 THEN LE=1
890 IF KS=13 THEN RI=1
900 IF KS=43 THEN UP=1
910 IF KS=49 THEN DN=1
920 IF KS=60 THEN FI=1
930 IF KS=62 THEN Q=1
940 RETURN
950 N$= " " : B$= " " : CHAR1, X, Y, B$: FORN=1TO3
960 GETKEYB$: IF VAL(B$)<0 OR VAL(B$)>9 THEN 960
970 N$=N$+B$: CHAR1, X, Y, N$: NEXT: CH=VAL(N$)
980 IF CH<96 OR CH>127 THEN 950

```

```
990 CH=CH-95:RETURN
1000 POKE65298,196:POKE65299,209:STOP
1010 RESTORE1020:FORN=0TO30:READA:POKE882+
N,A:NEXT:RETURN
1020 DATA160,0,132,212,132,214,169,208,133
,213,169,56,133,215,162,3
1030 DATA177,212,145,214,200,208,249,230,2
13,230,215,202,16,242,96
```

26

Music Monitor



Da Do RAM ROM

This three octave utility enables you to build up song sections and then to sequence them in harmony. Once produced in memory they can be saved, loaded, modified or just played for enjoyment.

Each sequence is built up first and then the song made is used to set the order in which each sequence is played. The program will prompt if you attempt to do things in the wrong order.

MENUs are used to display the options at each stage. If the option you require is not displayed pressing 'X' will exit to another menu. Selecting sequence mode at the beginning will produce the keyboard display.

The SEQUENCE CURSOR is moved left or right over the sequence boxes using the CBM KEY or the SHIFT KEY. The note required is then selected by using the cursor keys to move the note cursor. As the note cursor moves it will change to show the note played. The octave is indicated by the colour, sharps are 'reversed characters'.

When the note required is found then press space to enter the note into the sequence. The same note can then be placed elsewhere in that sequence by using the sequence cursor.

Selecting the playback mode will enable each sequence to be listened to. Pressing 'F' for faster and 'S' for slower will alter the tempo during playback.

A finished song can be saved to tape and loaded later from the appropriate menu. That way you can build up your own 'digital music' collection and if you are prone to bragging you can even tell all your friends that you have your own digital recording equipment!

```

10 REM >>> SEQUENTIAL CIRCUIT <<<
20 REM>>>DAVID<<<<<
30 VOL8
40 COLOR4,2,4:COLOR0,2,5:COLOR1,1:PRINTCHR
$(8)CHR$(142)
50 DIMN(35),NC(35),MU(7,1,15),AA(7,1,15):R
EM 8 SEQUENCES, 2 CHANNELS, 16 NOTES
60 FORR=0TO35:READN(R):NEXT
70 FORR=0TO35:READNC(R):NEXT
80 DB=3714:DP=12:SB=3436:SP=0:TEMPO=4:SI=0
::SO=0:REM SONG NOT IN
90 REM ****
100 PRINT"(CLR)":CHAR1,9,1,"{RVS ON}* SEQU
ENTIAL CIRCUIT *{RVS OFF}":COLOR1,10,2
110 CHAR1,11,4,"BY DAVID WHITTAKER"
120 COLOR1,1:CHAR1,11,10,"1. SONG MODE":C
HAR1,11,13,"2. SEQUENCE MODE"
130 GETA$:IFA$="1"THENGOSUB1090:GOTO100
140 IFA$="2"THENGOSUB170:GOTO100
150 GOTO130
160 REM ****
170 COLOR1,1:PRINT"(CLR)"SPC(12)"{RVS ON}
SEQUENCE MODE {RVS OFF}{CUR DN}{CUR DN}{C
UR DN}{CUR DN}{CUR DN}{CUR DN}""
180 FORR=1TO40:PRINT"(*P)":NEXT
190 PRINT"(HOME){CUR DN}{CUR DN}{CUR DN}{C
UR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{C
UR DN}{CUR DN}{CUR DN}{CUR RT}{CUR RT}"";
200 FORR=1TO36:PRINT"(*P)";
210 NEXT:PRINT
220 FORR=1TO5:PRINT"(CUR RT){CUR RT}"";
230 FORS=1TO3

```

```
240 PRINT"(*G){RVS ON} {RVS OFF} {RVS ON}
{RVS OFF} {RVS ON} {RVS OFF} {RVS ON}
{RVS OFF} {RVS ON} {RVS OFF} ";
250 NEXT
260 PRINT"(CUR L){*M}"
270 NEXT
280 FORR=1TO2:PRINT"(CUR RT){CUR RT}";
290 FORS=1TO3
300 PRINT"(*G){^H} {^G} {*G}{^H} {^B} {^G}
";
310 NEXT
320 PRINT"(CUR L){*M}"
330 NEXT
340 FORR=1TO39:PRINT"(*Y)";
350 NEXT:POKE4071,119
360 FORR=8TO23:POKE3072+R*40,101:POKE3072+
R*40+39,103:NEXT
370 PRINT"(HOME)"
380 CHAR1,37,9,"T"
390 CHAR1,36,10,"(^U){^*}{^I}"
400 FORR=11TO14:CHAR1,36,R,"(*Q){RED} {BLK
3}{*W}":NEXT
410 CHAR1,36,15,"(^J){^*}{^K}"
420 CHAR1,3,10,"{*A}"
430 FORR=4TO32STEP2:CHAR1,R,10,"(^*){*R}":
NEXT
440 CHAR1,34,10,"(^*){*S}"
450 CHAR1,1,11,"C1{^-}"
460 FORR=4TO34STEP2:CHAR1,R,11,"(^-)":NEX
T
470 CHAR1,3,12,"{*Q}"
480 FORR=4TO32STEP2:CHAR1,R,12,"(^*){^+}":
NEXT
490 CHAR1,34,12,"(^*){*W}"
500 CHAR1,1,13,"C2{^-}"
510 FORR=4TO34STEP2:CHAR1,R,13,"(^-)":NEX
T
520 CHAR1,3,14,"{*Z}"
530 FORR=4TO32STEP2:CHAR1,R,14,"(^*){*E}":
NEXT
540 CHAR1,34,14,"(^*){*X}"
550 GOSUB1020
560 FORR=12TO23:POKEDB-1024+R,37:POKEDB-10
24+12+R,113:NEXT
570 REM *****
```

```

580 CHAR1,9,3,"WHICH SEQUENCE 1 TO 8 ? "
590 GETA$:IFA$="X"THENRETURN
600 SQ=VAL(A$)-1:IFA$<"1"ORA$>"8"THEN590:E
LSEPRINTA$:GOSUB1800
610 CHAR1,11,3,"RECORD OR PLAYBACK ? "
620 GETM$:IFM$="X"THENRETURN
630 IFM$<>"R"ANDM$<>"P"THEN620:ELSEPRINTM$:
:GOSUB1800
640 IFM$="R"THENGOSUB670:CHAR1,0,0,"
":GOTO610
650 IFM$="P"THENGOSUB890:CHAR1,0,0,"
":GOTO610
660 REM RECORD MODE
670 CHAR1,0,0,"RECORD"
680 CHAR1,10,3,"CHANNEL 1 OR 2 ? "
690 GETA$:IFA$="X"THENGOSUB1800:RETURN
700 IFA$<"1"ORA$>"2"THEN690:ELSECH=VAL(A$)
-1:PRINTA$:GOSUB1800
710 POKE$B+SP,90:CHAR1,12,3,"JUST A MOMENT
...":GOSUB1820:GOSUB1800
720 GETA$:IFA$="X"THENRETURN
730 IFA$={CUR RT}"ANDDP<35THENPOKE$B+DP,1
11:DP=DP+1:SOUND1,N(DP),5
740 IFA$={CUR L}"ANDDP>0THENPOKE$B+DP,111
:DP=DP-1:SOUND1,N(DP),5
750 POKE$B+DP,NC(DP)
760 Z=PEEK(1347)
770 IFZ=2ANDSP>0 THENPOKE$B+SP,32:SP=SP-2
780 IFZ=1ANDSP<30THENPOKE$B+SP,32:SP=SP+2
790 NC=SP/2
800 POKE$B+SP,90
810 IFA$<>CHR$(13)THEN860
820 SI=1:MU(SQ,CH,NC)=N(DP)
830 X=NC(DP):CC=37:IFDP<12THENCC=0
840 IFDP>23THENCC=113
850 POKE$B+SP+80+CH*80,X:POKE2492+SP+CH*80
,CC:SOUND1,N(DP),5
860 IFA$=" "THENMU(SQ,CH,NC)=0:POKE$B+SP+8
0+CH*80,46:SOUND1,0,5
870 GOTO720
880 REM PLAYBACK MODE
890 IFSI=0THENCHAR1,13,3,"NONE IN MEMORY!"
:FORT=1TO1000:NEXT:GOSUB1800:RETURN
900 CHAR1,0,0,"PLAYBACK":P=0
910 GETA$:IFA$="X"THENRETURN

```

```

920 GOSUB990
930 M=MU(SQ,0,P):IFMTHEN SOUND1,M,TEMPO*3
940 N=MU(SQ,1,P):IFNTHEN SOUND2,N,TEMPO*3
950 IFM+N=0 THEN FORT=1 TO TEMPO*50:NEXT
960 P=P+1:IFP=16 THEN P=0
970 GOTO910
980 REM UPDATE TEMPO
990 IFA$="F" THEN IF TEMPO>0 THEN TEMPO=TEMPO-1
:GOTO1020
1000 IFA$="S" THEN IF TEMPO<8 THEN TEMPO=TEMPO+
1:GOTO1020
1010 RETURN
1020 FORR=3549 TO 3629 STEP 40:POKER,32:NEXT
1030 POKE3669,248:IF TEMPO>1 THEN POKE3669,16
0
1040 IF TEMPO>2 THEN POKE3629,248:IF TEMPO>3 TH
EN POKE3629,160
1050 IF TEMPO>4 THEN POKE3589,248:IF TEMPO>5 TH
EN POKE3589,160
1060 IF TEMPO>6 THEN POKE3549,248:IF TEMPO>7 TH
EN POKE3549,160
1070 RETURN
1080 REM OPENING SCREEN
1090 PRINT"(CLR)":COLOR1,1
1100 CHAR0,14,0,"{RVS ON} SONG MODE {RVS
OFF}"
1110 CHAR1,14,7,"P PLAY SONG"
1120 CHAR1,14,9,"L LOAD SONG"
1130 CHAR1,14,11,"S SAVE SONG"
1140 CHAR1,14,13,"E EDIT SONG"
1150 CHAR1,14,16,"X EXIT MENU"
1160 GETA$:IFA$="X" THEN RETURN
1170 IFA$="P" THEN GOSUB1230:GOTO1090
1180 IFA$="L" THEN GOSUB1340:GOTO1090
1190 IFA$="S" THEN GOSUB1510:GOTO1090
1200 IFA$="E" THEN GOSUB1650:GOTO1090
1210 GOTO1160
1220 REM PLAY SONG
1230 IFS0=0 THEN PRINT"(CLR){CUR DN}{CUR RT}
NO SONG IN MEMORY !":FORT=1 TO 1000:NEXT:RET
URN
1240 PRINT"(CLR)PLAY SONG MODE{CUR DN}{CUR
DN}{CUR DN}{CUR DN}{CUR DN}X TO EXIT"
1250 P=0:Q=0
1260 GETA$:IFA$="X" THEN RETURN

```

```
1270 GOSUB990
1280 M=MU(P,0,Q):IFMTHEN SOUND1,M,TEMPO*3
1290 N=MU(P,1,Q):IFNTHEN SOUND2,N,TEMPO*3
1300 IFM+N=0 THEN FORT=1 TO TEMPO*50:NEXT
1310 Q=Q+1:IFQ=16 THEN Q=0:P=P+1:IFP=8 THEN 12
50
1320 GOTO1260
1330 REM LOAD SONG
1340 IFSI=0 THEN 1380
1350 PRINT"(CLR)":CHAR1,5,5,"OVERWRITE CUR
RENT SONG Y/N ?"
1360 GETA$:IFA$="N" THEN RETURN
1370 IFA$<>"Y" THEN 1360
1380 PRINT"(CLR)LOAD MODE"
1390 CHAR1,1,5," ":INPUT"FILENAME (MAX 8 C
HARS)":FI$
1400 IFLEN(FI$)<1 OR LEN(FI$)>8 THEN 1390
1410 IFFI$="X" THEN RETURN
1420 OPEN1,1,0,FI$
1430 FORR=0 TO 7
1440 FORS=0 TO 1
1450 FORT=0 TO 15
1460 INPUT#1,MU(R,S,T)
1470 NEXTT,S,R:CLOSE1:SI=1:SO=1
1480 RETURN
1490 PRINT"(CUR DN)(CUR RT)DONE.":FORT=1 TO
1000:NEXT
1500 REM SAVE SONG
1510 IFSI=0 THEN PRINT"(CLR)(CUR DN)(CUR RT)
NO SONG IN MEMORY !":FORT=1 TO 1000:NEXT:RET
URN
1520 PRINT"(CLR)SAVE MODE"
1530 CHAR1,1,5," ":INPUT"FILENAME (MAX 8 C
HARS)":FI$
1540 IFLEN(FI$)<1 OR LEN(FI$)>8 THEN 1530
1550 IFFI$="X" THEN RETURN
1560 OPEN1,1,2,FI$
1570 FORR=0 TO 7
1580 FORS=0 TO 1
1590 FORT=0 TO 15
1600 PRINT#1,MU(R,S,T)
1610 NEXTT,S,R:CLOSE1
1620 PRINT"(CUR DN)(CUR RT)DONE.":FORT=1 TO
1000:NEXT
1630 RETURN
```

```
1640 REM EDIT SONG
1650 IFSI=0THENPRINT"(CLR){CUR DN}{CUR RT}
NO SEQUENCES IN MEMORY !":FORT=1TO1000:NEXT:RETURN
1660 PRINT"(CLR)EDIT MODE"
1670 CHAR1,1,5,"KEY IN ANY 8 SEQUENCES TO
MAKE A SONG":S0=1
1680 FORR=0TO7:CHAR1,12+R*2,10,"(*T)":NEXT
1690 FORR=0TO7:A$=""
1700 GETA$:IFA$<"1"ORA$>"8"THEN1700:ELSEA=
VAL(A$)-1
1710 CHAR1,12+R*2,9,A$
1720 FORS=0TO15:AA(R,0,S)=MU(A,0,S):AA(R,1,
,S)=MU(A,1,S)
1730 NEXTS,R
1740 PRINT:PRINT"(CUR DN){CUR DN}{CUR RT}W
AIT A MOMENT..."
1750 FORR=0TO7:FORS=0TO1:FORT=0TO15
1760 MU(R,S,T)=AA(R,S,T)
1770 NEXTT,S,R
1780 PRINT"(CUR DN){CUR DN}{CUR RT}DONE.":FORT=1TO1000:NEXT
1790 RETURN
1800 FORT=1TO200:NEXT:FORT=0TO38:CHAR1,T,3
,:NEXT:RETURN
1810 REM DISPLAY
1820 FORR=0TO15
1830 X=MU(SQ,0,R):IFX=0THENX=46:CC=1:ELSEG
OSUB1960
1840 POKE3516+R*2,X:POKE2492+R*2,CC
1850 X=MU(SQ,1,R):IFX=0THENX=46:CC=1:ELSEG
OSUB1960
1860 POKE3596+R*2,X:POKE2572+R*2,CC
1870 NEXT:RETURN
1880 REM ****
1890 DATA169,205,262,303,345,383,425,453,4
85,516,545,571
1900 DATA596,619,643,667,685,704,723,739,7
55,770,785,798
1910 DATA810,822,834,844,854,864,873,881,8
89,897,904,911
1920 REM****
1930 DATA3,131,4,132,5,6,134,7,135,1,129,2
1940 DATA3,131,4,132,5,6,134,7,135,1,129,2
1950 DATA3,131,4,132,5,6,134,7,135,1,129,2
```

```
1960 RR=0
1970 IFN(RR)=XTHEN1980:ELSEERR=RR+1:GOT0197
0
1980 X=NC(RR):IFRR<12THENCC=0:RETURN
1990 IFRR>23THENCC=113:RETURN
2000 CC=37:RETURN
```

Final Words

In producing this book an attempt has been made to adopt a standard of game which is well above the norm.

This naturally leads to longer programs which require greater care to enter. The time will come therefore when you will need to debug your programs. This is when you have the opportunity to try to understand how and why the programs work.

To help you in this task there follows a ‘debug guide’.

General points

1. Get someone else to double check that the entry matches the listing. Say it out loud.
2. Double check any special characters. Reread the chapter entitled *Important Entry Notes*.
3. Be careful that an I has not been confused with a 1.
4. Ensure that lower case ‘a’ is entered as such and not ‘A’.
5. Watch out for ‘;’ instead of ‘:’.
6. Double check spaces by reference to the spaces above and below.
7. Double check line numbers. Don’t confuse a number which flows over onto another line as a line number.
8. Treble check data. It is very easy to skip a few numbers. As an overall check, count how many commas and compare with the listing.
9. Make sure that all quotes are in.
10. Is there really a fault? Reread the instructions for play. Most responses need to be followed by the RETURN key.

Specific problems

1. Special characters have been used for listing. Familiarise yourself with these, on a simple program first. We anticipate some problems from readers in using this technique, but consider the finished results to be worth while for the majority.
2. One of the main sources of program failure is the handling of bad entries. For example, a program may crash if an unexpected minus number were entered. It is possible to anticipate most 'silly' entries but only at the expense of needlessly longer listings. In writing these programs it has been assumed that the player will not set out to crash the program! They are therefore only protected against class three idiots. If you will be exposing your finished efforts to a hostile environment then it is up to you to protect against class one idiots.
3. All programs should work equally well on monochrome sets, but change the colours (after saving the correct version) to suit your own taste.
4. It is always good practice to load into a machine which has just been powered up. This can avoid problems if the previous program left something changed deep within the machine.
5. If a program crashes producing a weird screen, type in the following (be careful because what you type will not be seen on screen).

POKE 65298, 196: POKE 65299, 209 [RETURN]

The screen should return to normal, with an error message. Pressing 'help' will then highlight the problem.

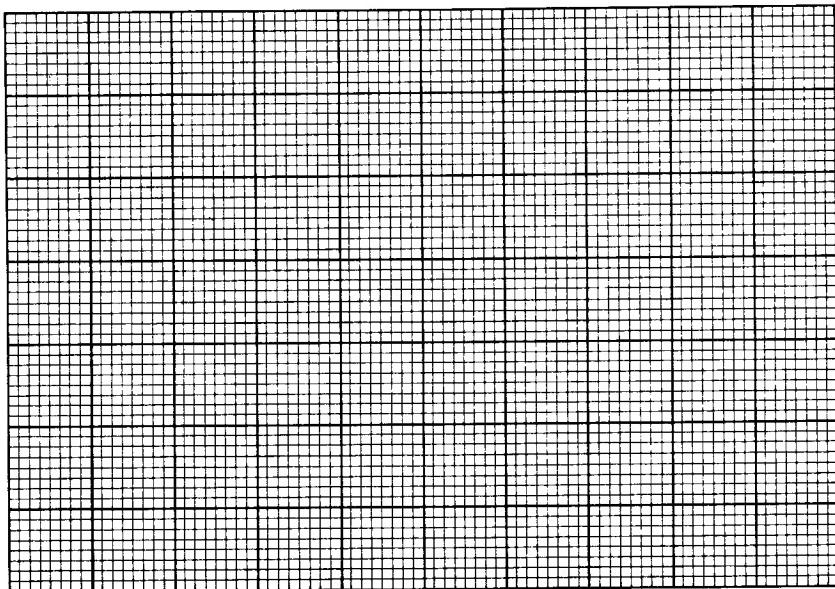
Debugging

If none of the foregoing resolve the problem then roll up your sleeves and note some of these tips.

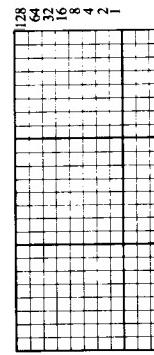
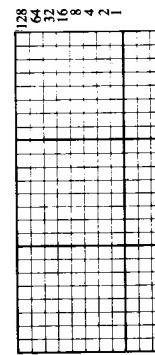
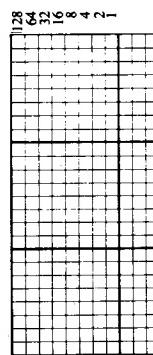
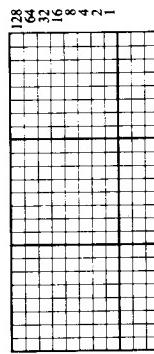
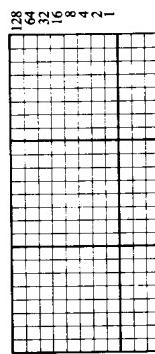
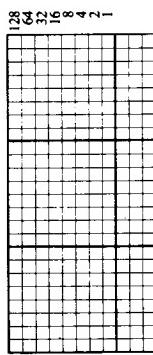
1. If an error message indicates a fault is in a particular line, but it looks right when examined, remember that the fault could be in the line which passes information to that line. Check back related lines.
2. Errors such as 'Out of range', 'Improper argument', etc., may be traced as above. To help type 'PRINT A' or whatever variables are used in that line. This should indicate the problem.
3. Bits being missed out! The 'TRON' function will show the order in which lines are called, often highlighting a rogue 'GOSUB' or similar problem.

4. Work methodically. Consider each section in turn. Set ‘Break points’ using ‘End’ or dump out variables by adding a short print line in the middle of the faulty routine.
5. Temporarily disable sections of program by adding a ‘REM’ to the start of a line. See what the effect is.
6. Re-type the section responsible!
7. Use the ‘TRAP’ function as described in the manual.
8. To RESET without losing the program, first hold down the RUN/STOP key and then whilst held, press the RESET.
The machine will then initialise to the built in machine code monitor. To return to BASIC simply type ‘X’ [RETURN] and you can list to correct any errors before running again.
9. Go to bed – have a sleep. It sometimes works the next day!

Appendix 1: Defined Character Design Aid



CHARACTER DEFINITIONS



You may photocopy this page for your own use.

Appendix 2: Screen Layout Planner

40 COLUMNS x 25 ROWS

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

You may photocopy this page for your own use.

